IWCS 2025

16th International Conference on Computational Semantics

Proceedings of the Conference

September 22-23, 2025

©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 979-8-89176-316-6

Introduction

These are the proceedings of the 16h International Conference on Computational Semantics, held at Heinrich Heine University, Düsseldorf, Germany, from 22 to 24 September 2025.
The aim of IWCS is to bring together researchers interested in all areas of computational semantics and computational aspects of meaning of natural language within written, spoken, signed, or multimodal communication. As shown in these proceedings, covered topics embrace both symbolic and machine learning approaches to computational semantics, in relation with multimodality, external knowledge, cognitive analysis, and many resources, e.g., annotation and software. We invited three keynote speakers to present their work: Oana-Maria Camburu (Department of Computing, Imperial College London, UK), Alexander Koller (Department of Language Science and Technology, Saarland University, Germany), and Denis Paperno (Utrecht University, Netherlands). The program also includes five oral presentation sessions and two poster sessions. Three satellite workshops will be held on the day after the conference:

- Bridges and Gaps between Formal and Computational Linguistics (BriGap2);
- The Second International Workshop on Construction Grammars and Natural Language Processing (CxGs+NLP 2025);
- A workshop joining the Second Workshop on Multimodal Semantic Representations (MMSR II) and the 21st Joint ACL – ISO Workshop on Interoperable Semantic Annotation (ISA-21).

We received 50 submissions (39 long and 11 short submissions) that each was assigned three reviewers. One long submission was withdrawn before review. Out of the remaining ones, 31 papers were accepted for the conference (25 long and 6 short), resulting in 14 oral presentations and 18 poster presentations, and a final acceptance rate of 63% (66% for long papers and 55% for short papers). Twp long paper were withdrawn after acceptance. We are very grateful to the reviewers for their work and discussions that allowed us to produce a high-quality program for the conference.
In addition to this scientific work, this conference was made possible thanks to the local organizing team and the support of Haus der Universität.
We hope that IWCS 2025 will be an exciting edition and a lively forum to the computational semantics community.

Kilian Evang, Laura Kallmeyer, and Sylvain Pogodalla
September 2025

Organizing Committee

Program Chairs

Kilian Evang, Heinrich Heine University Düsseldorf
Laura Kallmeyer, Heinrich Heine University Düsseldorf
Sylvain Pogodalla, LORIA/Inria Nancy

Local Organizers

Long Chen, Heinrich Heine University Düsseldorf
Rafael Ehren, Heinrich Heine University Düsseldorf
Kilian Evang, Heinrich Heine University Düsseldorf
Laura Kallmeyer, Heinrich Heine University Düsseldorf
Rainer Osswald, Heinrich Heine University Düsseldorf
Christian Wurm, Heinrich Heine University Düsseldorf
Deniz Ekin Yavaş, Heinrich Heine University Düsseldorf

Program Committee

Program Committee

Rodrigo Agerri, University of the Basque Country
Maxime Amblard, Université de Lorraine
Daisuke Bekki, Ochanomizu University
Moritz Blum, Universität Bielefeld
Claire Bonial, Georgetown University
Johan Bos, University of Groningen
Susan Windisch Brown, University of Colorado at Boulder
Harry Bunt, Tilburg University
Aljoscha Burchardt, German Research Center for AI
Miriam Butt, Universität Konstanz
Long Chen, Heinrich-Heine Universität Düsseldorf
Emmanuele Chersoni, The Hong Kong Polytechnic University
Philipp Cimiano, Bielefeld University
Robin Cooper, University of Gothenburg
Marie Cousin, Université de Lorraine
Rodolfo Delmonte, Monash University
Markus Egg, Humboldt-Universität zu Berlin
Katrin Erk, University of Texas
Neele Falk, University of Stuttgart
Pascale Feldkamp, School of Communication and Culture
Francis Ferraro, University of Maryland
James Fodor, University of Melbourne
Philippe de Groote, INRIA
Bruno Guillaume, INRIA
Simon Guillot, Universite du Maine
Dag Trygve Truslew Haug, University of Oslo
Johannes Heinecke, Orange-labs
Elisabetta Jezek, University of Pavia
Rohit J. Kate, University of Wisconsin - Milwaukee
Gene Louis Kim, University of South Florida
Philippe Langlais, Université de Montréal
Staffan Larsson, Göteborg University
Chuyuan Li, University of British Columbia
Zhaohui Luo, Royal Holloway University of London
Vladislav Maraev, Göteborg University
Gaëtan Margueritte, Tokyo University
Philipp Meier, Heinrich-Heine Universität Düsseldorf
Timothee Mickus, University of Helsinki
Yusuke Miyao, The University of Tokyo
Richard Moot, CNRS
Lawrence S. Moss, Indiana University at Bloomington
Dmitry Nikolaev, University of Manchester
Bill Noble, Göteborg University
Rik van Noord, University of Groningen
Juri Opitz, University of Zurich
Martha Palmer, University of Colorado at Boulder

Violaine Prince, University of Montpelier
Stephen Pulman, Apple
Matthew Purver, Queen Mary University of London
Giulia Rambelli, University of Bologna
Valentin D. Richard, University of Amsterdam
Kyeongmin Rim, Brandeis University
Mehrnoosh Sadrzadeh, University College London
Asad B. Sayeed, University of Gothenburg
Sabine Schulte im Walde, University of Stuttgart
Ravi Shekhar, University of Essex
Mollie Shichman, DEVCOM Army Research Laboratory
Mark Steedman, University of Edinburgh
Bonnie Webber, Edinburgh University
Shira Wein, Amherst College
Hitomi Yanaka, the University of Tokyo
Deniz Ekin Yavas, Heinrich Heine University Duesseldorf
Bingyang Ye, Brandeis University
Annie Zaenen, Stanford University
Alessandra Zarcone, Technische Hochschule Augsburg
Sina Zarrieß, Bielefeld University

Keynote Talk

Thoughts You Can Trust? Evaluating the Faithfulness of Model-Generated Explanations and Their Effects on Human Performance

Oana-Maria Camburu
Imperial College London

Abstract: Large Language Models (LLMs) can readily generate natural language explanations—or chain-of-thoughts (CoTs)—to justify their outputs. In this talk, I will first introduce methods for evaluating whether such explanations faithfully reflect the decision-making processes of the models that produce them. Second, I present the results of a user study involving 85 clinicians and medical students diagnosing chest X-rays. The study compares the effectiveness of natural language explanations, saliency maps, and their combination in supporting clinical decision-making.

Bio: Oana-Maria Camburu is an Assistant Professor in the Department of Computing at Imperial College London. Prior to that, she was a Principal Research Fellow in the Department of Computer Science at the University College London, holding an Early Career Leverhulme Fellowship. Oana was also a postdoc at the University of Oxford, from where she obtained her PhD in "Explaining Deep Neural Networks". Her main research interests lie in explainability for deep learning models and AI safety and alignment.

Keynote Talk

Compositionality, Intensionality and LLMs: The Case of the Personal Relations Task

Denis Paperno
Utrecht University

Abstract: Semanticists have long considered compositionality to be at the heart of natural language interpretation. Modern large language models (LLMs) achieve impressive results at tasks involving semantics, but in most cases it is hard to judge to what extent they rely on compositional mechanisms. Since the training data is enormous and could contain many complex phrases, much of LLM's performance could potentially be attributed to non-compositional pattern memorization, leaving little space for compositional ability. For example, "mother of Elon Musk" could be processed by a language model as a holistic unit since the phrase occurs in this form in the training corpora. The talk will discuss ongoing work based on the Personal Relations task (Paperno, 2022), designed to assess semantic compositionality properly. The Personal Relations task relies on a small universe with randomly generated relations which can not be present in language model pretraining, therefore offering a testing ground for compositional abilities of models at phrase level. Furthermore, the Personal Relations task allows us to contrast intensional and extensional semantic interpretation. We find that language models (still) exhibit different compositional abilities than humans, with intensionality playing a substantial role.

Denis Paperno. 2022. On Learning Interpreted Languages with Recurrent Models. *Computational Linguistics*, 48(2):471–482.

Bio: Denis Paperno is assistant professor of computational linguistics at Utrecht University. He received a PhD in Linguistics from the University of California Los Angeles, and subsequently worked at the University of Trento (CLIC lab, Rovereto) as a postdoc and at the Loria lab (Nancy) as a CNRS researcher. Denis has published extensively in the fields of semantics, language model evaluation, and vector space representations of meaning. His research contributions include work on compositionality in computational models of semantics, visual grounding, and representation probing.

Keynote Talk

Solving Complex Problems with Large Language Models

Alexander Koller
Saarland University

Abstract: One of the great promises that people connect with LLMs is that they can make complex problem-solving with computers accessible to lay users. Unlike traditional symbolic solvers (e.g. for planning or constraint solving), LLMs accept natural-language input and require no expert training; unlike earlier task-oriented dialogue systems, they can be applied across arbitrary domains. In my talk, I will explore the degree to which LLMs are fulfilling this promise. I will present recent work on whether current LLMs "reason" or "recite", discuss the role of symbolic representations in LLM-based problem-solving, and offer some thoughts on trustworthy problem-solving with LLMs.

Bio: Alexander Koller is a Professor of Computational Linguistics at Saarland University in Saarbrücken, Germany. His research interests include planning and reasoning with LLMs, syntactic and semantic processing, natural language generation, and dialogue systems. He is particularly interested in neuro-symbolic models that bring together principled linguistic modeling and correctness guarantees with the coverage and robustness of neural approaches. Alexander received his PhD from Saarland University and was previously a postdoc at Columbia University and the University of Edinburgh, faculty at the University of Potsdam, and Visiting Senior Research Scientist at the Allen Institute for AI.

Table of Contents

Advancing the Database of Cross-Linguistic Colexifications with New Workflows and Data
Annika Tjuka, Robert Forkel, Christoph Rzymski and Johann-Mattis List 1

FRIDA to the Rescue! Analyzing Synthetic Data Effectiveness in Object-Based Common Sense Reasoning for Disaster Response
Mollie Shichman, Claire Bonial, Austin Blodgett, Taylor Pellegrin, Francis Ferraro and Rachel Rudinger 16

SemToken: Semantic-Aware Tokenization for Efficient Long-Context Language Modeling
Dong Liu and Yanxuan Yu 30

ding-01 :ARG0: An AMR Corpus for Spontaneous French Dialogue
Jeongwoo Kang, Maria Boritchev and Maximin Coavoux 40

A Graph Autoencoder Approach for Gesture Classification with Gesture AMR
Huma Jamil, Ibrahim Khebour, Kenneth Lai, James Pustejovsky and Nikhil Krishnaswamy . . . 51

Retrieval-Augmented Semantic Parsing: Improving Generalization with Lexical Knowledge
Xiao Zhang, Qianru Meng and Johan Bos 59

Not Just Who or What: Modeling the Interaction of Linguistic and Annotator Variation in Hateful Word Interpretation
Sanne Hoeken, Özge Alacam, Dong Nguyen, Massimo Poesio and Sina Zarrieß 73

Context Effects on the Interpretation of Complement Coercion: A Comparative Study with Language Models in Norwegian
Matteo Radaelli, Emmanuele Chersoni, Alessandro Lenci and Giosuè Baggio 88

LLMs Struggle with NLI for Perfect Aspect: A Cross-Linguistic Study in Chinese and Japanese
LU Jie, Du Jin and Hitomi Yanaka 99

Assessing LLMs' Understanding of Structural Contrasts in the Lexicon
Shuxu LI, Antoine Venant, Philippe Langlais and François Lareau 108

A German WSC dataset comparing coreference resolution by humans and machines
Wiebke Petersen and Katharina Spalek 120

Finding Answers to Questions: Bridging between Type-based and Computational Neuroscience Approaches
Staffan Larsson, Jonathan Ginzburg, Robin Cooper and Andy Lücking 128

Can Large Language Models Robustly Perform Natural Language Inference for Japanese Comparatives?
Yosuke Mikami, Daiki Matsuoka and Hitomi Yanaka 137

Is neural semantic parsing good at ellipsis resolution, or isn't it?
Xiao Zhang and Johan Bos 147

Extracting Behaviors from German Clinical Interviews in Support of Autism Spectrum Diagnosis
Margareta A. Kulcsar, Ian Paul Grant and Massimo Poesio 153

The Proper Treatment of Verbal Idioms in German Discourse Representation Structure Parsing
Kilian Evang, Rafael Ehren and Laura Kallmeyer 166

Does discourse structure help action prediction? A look at Correction Triangles.
Kate Thompson, Akshay Chaturvedi and Nicholas Asher 176

FAMWA: A new taxonomy for classifying word associations (which humans improve at but LLMs still struggle with)
Maria A. Rodriguez, Marie Candito and Richard Huyghe 185

Computational Semantics Tools for Glue Semantics
Mark-Matthias Zymla, Mary Dalrymple and Agnieszka Patejuk 199

Which Model Mimics Human Mental Lexicon Better? A Comparative Study of Word Embedding and Generative Models
Huacheng Song, Zhaoxin Feng, Emmanuele Chersoni and Chu-Ren Huang 218

Semantic Analysis Experiments for French Citizens' Contribution : Combinations of Language Models and Community Detection Algorithms
Sami Guembour, Dominguès Dominguès and Sabine Ploux 241

Neurosymbolic AI for Natural Language Inference in French : combining LLMs and theorem provers for semantic parsing and natural language reasoning
Maximos Skandalis, Lasha Abzianidze, Richard Moot, Christian Retoré and Simon Robillard 252

ProPara-CRTS: Canonical Referent Tracking for Reliable Evaluation of Entity State Tracking in Process Narratives
Bingyang Ye, Timothy Obiso, Jingxuan Tu and James Pustejovsky 264

The Difficult Case of Intended and Perceived Sarcasm: a Challenge for Humans and Large Language Models
Hyewon Jang and Diego Frassinelli 279

A Model of Information State in Situated Multimodal Dialogue
Kenneth Lai, Lucia Donatelli, Richard Brutti and James Pustejovsky 292

Learning to Refer: How Scene Complexity Affects Emergent Communication in Neural Agents
Dominik Künkele and Simon Dobnik 299

On the Role of Linguistic Features in LLM Performance on Theory of Mind Tasks
Ekaterina Kozachenko, Gonçalo Guiomar and Karolina Stanczak 308

Mapping Semantic Domains Across India's Social Media: Networks, Geography, and Social Factors
Gunjan Anand and Jonathan Dunn 317

Disentangling lexical and grammatical information in word embeddings
Li Liu and François Lareau 331

Program

Haus der Universität, Schadowplatz 14, 40212 Düsseldorf

Monday, September 22, 2025

08:30 - 09:15 *Registration (Level 0)*

09:15 - 09:30 *Opening remarks (Level -1)*

09:30 - 10:30 *Invited speaker: Oana-Maria Camburu (Level -1)*

10:30 - 11:00 *Coffee break (Level 0)*

11:00 - 12:30 *Main session 1 (semantic parsing, linguistic phenomena) (Level -1)*

Neurosymbolic AI for Natural Language Inference in French : combining LLMs and theorem provers for semantic parsing and natural language reasoning
Maximos Skandalis, Lasha Abzianidze, Richard Moot, Christian Retoré and Simon Robillard

Is neural semantic parsing good at ellipsis resolution, or isn't it?
Xiao Zhang and Johan Bos

Retrieval-Augmented Semantic Parsing: Improving Generalization with Lexical Knowledge
Xiao Zhang, Qianru Meng and Johan Bos

12:30 - 14:00 *Lunch break*

14:00 - 15:00 *Invited speaker: Denis Paperno (Level -1)*

15:00 - 16:30 *Coffee break (Level 0)*

15:00 - 16:30 *Poster session 1a (LLMs) (Level 2)*

Can Large Language Models Robustly Perform Natural Language Inference for Japanese Comparatives?
Yosuke Mikami, Daiki Matsuoka and Hitomi Yanaka

LLMs Struggle with NLI for Perfect Aspect: A Cross-Linguistic Study in Chinese and Japanese
LU Jie, Du Jin and Hitomi Yanaka

Monday, September 22, 2025 (continued)

Assessing LLMs' Understanding of Structural Contrasts in the Lexicon
Shuxu LI, Antoine Venant, Philippe Langlais and François Lareau

The Difficult Case of Intended and Perceived Sarcasm: a Challenge for Humans and Large Language Models
Hyewon Jang and Diego Frassinelli

On the Role of Linguistic Features in LLM Performance on Theory of Mind Tasks
Ekaterina Kozachenko, Gonçalo Guiomar and Karolina Stanczak

15:00 - 16:30 *Poster session 1b (sociolinguistics, applied linguistics) (Level 3)*

Extracting Behaviors from German Clinical Interviews in Support of Autism Spectrum Diagnosis
Margareta A. Kulcsar, Ian Paul Grant and Massimo Poesio

Semantic Analysis Experiments for French Citizens' Contribution : Combinations of Language Models and Community Detection Algorithms
Sami Guembour, Dominguès Dominguès and Sabine Ploux

Mapping Semantic Domains Across India's Social Media: Networks, Geography, and Social Factors
Gunjan Anand and Jonathan Dunn

FRIDA to the Rescue! Analyzing Synthetic Data Effectiveness in Object-Based Common Sense Reasoning for Disaster Response
Mollie Shichman, Claire Bonial, Austin Blodgett, Taylor Pellegrin, Francis Ferraro and Rachel Rudinger

16:30 - 18:00 *Main session 2 (LLMs and compositionality, hybrid approaches) (Level -1)*

Context Effects on the Interpretation of Complement Coercion: A Comparative Study with Language Models in Norwegian
Matteo Radaelli, Emmanuele Chersoni, Alessandro Lenci and Giosuè Baggio

19:00 - 21:00 *Conference dinner (ALEX Düsseldorf, Graf-Adolf-Platz 15, 40213 Düsseldorf)*

Tuesday, September 23, 2025

08:30 - 09:30 *Registration (Level 0)*

09:30 - 10:30 *Invited speaker: Alexander Koller (Level -1)*

10:30 - 11:00 *Coffee break (Level 0)*

11:00 - 12:30 *Main session 3 (language models and linguistic knowledge) (Level -1)*

Disentangling lexical and grammatical information in word embeddings
Li Liu and François Lareau

The Proper Treatment of Verbal Idioms in German Discourse Representation Structure Parsing
Kilian Evang, Rafael Ehren and Laura Kallmeyer

SemToken: Semantic-Aware Tokenization for Efficient Long-Context Language Modeling
Dong Liu and Yanxuan Yu

12:30 - 14:00 *Lunch break*

14:00 - 15:00 *Main session 4 (semantics and cognition) (Level -1)*

Which Model Mimics Human Mental Lexicon Better? A Comparative Study of Word Embedding and Generative Models
Huacheng Song, Zhaoxin Feng, Emmanuele Chersoni and Chu-Ren Huang

Finding Answers to Questions: Bridging between Type-based and Computational Neuroscience Approaches
Staffan Larsson, Jonathan Ginzburg, Robin Cooper and Andy Lücking

15:00 - 16:30 *Coffee break (Level 0)*

15:00 - 16:30 *Poster session 2a (resources) (Level 2)*

A German WSC dataset comparing coreference resolution by humans and machines
Wiebke Petersen and Katharina Spalek

FAMWA: A new taxonomy for classifying word associations (which humans improve at but LLMs still struggle with)
Maria A. Rodriguez, Marie Candito and Richard Huyghe

Advancing the Database of Cross-Linguistic Colexifications with New Workflows and Data
Annika Tjuka, Robert Forkel, Christoph Rzymski and Johann-Mattis List

ding-01 :ARG0: An AMR Corpus for Spontaneous French Dialogue
Jeongwoo Kang, Maria Boritchev and Maximin Coavoux

Computational Semantics Tools for Glue Semantics
Mark-Matthias Zymla, Mary Dalrymple and Agnieszka Patejuk

15:00 - 16:30 *Poster session 2b (discourse and context) (Level 3)*

ProPara-CRTS: Canonical Referent Tracking for Reliable Evaluation of Entity State Tracking in Process Narratives
Bingyang Ye, Timothy Obiso, Jingxuan Tu and James Pustejovsky

Does discourse structure help action prediction? A look at Correction Triangles.
Kate Thompson, Akshay Chaturvedi and Nicholas Asher

Learning to Refer: How Scene Complexity Affects Emergent Communication in Neural Agents
Dominik Künkele and Simon Dobnik

A Graph Autoencoder Approach for Gesture Classification with Gesture AMR
Huma Jamil, Ibrahim Khebour, Kenneth Lai, James Pustejovsky and Nikhil Krishnaswamy

16:30 - 17:30 *Main session 5 (situated interpretation) (Level -1)*

Not Just Who or What: Modeling the Interaction of Linguistic and Annotator Variation in Hateful Word Interpretation
Sanne Hoeken, Özge Alacam, Dong Nguyen, Massimo Poesio and Sina Zarrieß

A Model of Information State in Situated Multimodal Dialogue
Kenneth Lai, Lucia Donatelli, Richard Brutti and James Pustejovsky

Tuesday, September 23, 2025 (continued)

17:30 - 18:00 *Closing remarks (Level -1)*

Wednesday, September 24, 2025

08:30 - 09:00 *Registration (Level 0)*

09:00 - 18:00 *The Second International Workshop on Construction Grammars and Natural Language Processing (CxGs+NLP 2025) (Level -1)*

09:00 - 18:00 *ISA-21, The 21st Joint ACL - ISO Workshop on Interoperable Semantic Annotation (Level 2)*

09:00 - 18:00 *Bridges and Gaps between Formal and Computational Linguistics (BriGap2) (Level 3)*

10:30 - 11:00 *Coffee break (Level 0)*

15:30 - 16:00 *Coffee break (Level 0)*

Advancing the Database of Cross-Linguistic Colexifications with New Workflows and Data

Annika Tjuka[1], Robert Forkel[1,2], Christoph Rzymski[1], Johann-Mattis List[2,1]

[1]Max Planck Institute for Evolutionary Anthropology, Leipzig, Germany
[2]Chair of Multilingual Computational Linguistics, University of Passau, Passau, Germany
Correspondence: annika_tjuka@eva.mpg.de and mattis.list@uni-passau.de

Abstract

Lexical resources are crucial for cross-linguistic analysis and can provide new insights into computational models for natural language learning. Here, we present an advanced database for comparative studies of words with multiple meanings, a phenomenon known as colexification. The new version includes improvements in the handling, selection and presentation of the data. We compare the new database with previous versions and find that our improvements provide a more balanced sample covering more language families worldwide, with enhanced data quality, given that all word forms are provided in phonetic transcription. We conclude that the new Database of Cross-Linguistic Colexifications has the potential to inspire exciting new studies that link cross-linguistic data to open questions in linguistic typology, historical linguistics, psycholinguistics, and computational linguistics.

1 Introduction

The *Database of Cross-Linguistic Colexifications* (CLICS, https://clics.clld.org, Rzymski et al., 2020) offers detailed data on the distribution and frequency of *colexifications* across several thousand languages. Colexification is a cover term that unifies the notions of polysemy, homophony, and underspecification, referring to cases where a single word form in a given language expresses multiple senses (François, 2008). For example, Vietnamese *xanh* refers to 'blue' and 'green' at the same time, German *böse* means both 'angry' and 'evil', or English *ear* refers to a part of the body or a part of a grain. The different examples represent words with multiple senses and can be labeled as underspecification (Vietnamese), polysemy (German), or homophony (English), but they can also be taken together as examples of the phenomenon of colexification.

CLICS has built on this idea by collecting data from multilingual word lists that were unified with respect to the semantic glosses by which words across different languages are elicited. From these word lists, colexifications were automatically extracted, forming a large *colexification network* (List et al., 2013) that can be investigated interactively (Mayer et al., 2014). The database has improved concerning the workflow by which data are aggregated and in terms of the number of datasets underlying the database (4 datasets in Version 1.0, List et al. 2014, 15 datasets in Version 2.0, List et al. 2018, 30 datasets in Version 3.0 Rzymski et al. 2020). In its current form, the CLICS database is characterized by three major features. First, CLICS *aggregates* data from existing standardized datasets, rather than curating data directly. Second, CLICS offers its data in both *machine- and human-readable form*, allowing scholars to access the data in computational workflows as well as through the web interface. Third, CLICS is *open*, and both the individual data and the source code are published with permissive licenses, allowing scholars not only to investigate the database, but also to extend it with additional content or methods.

Given that five years have passed since the last official release of CLICS and that new relevant datasets have been published during this time, mainly as part of Lexibank, a large repository for standardized multilingual word lists (https://lexibank.clld.org, List et al., 2022; Blum et al., 2025), it is time to improve the database even further. Taking advantage of the fact that CLICS is open and free to modification, we therefore present an updated version of the CLICS database, which we named CLICS 4 for convenience. CLICS 4 not only increases the underlying data, but also addresses three major shortcomings of the previous versions of CLICS by improving (1) the handling of concepts (§ 3.2), (2) the selection of languages to be included in the colexification database (§ 3.3),

Proceedings of the 16th International Conference on Computational Semantics, pages 1–15
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

	forest	*tree*	*wood*
French	fɔʀɛ, bwɑ	aʀbrə	bwɑ
Russian	lʲes	dʲerɪva	dʲerɪva
Yukaghir	aːnmonilʲe	saːl	saːl
Yaqui	dʒuja	dʒuja	kuta

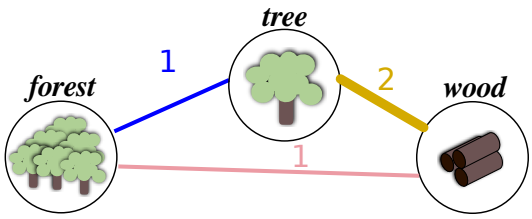


Figure 1: Cross-linguistic colexifications (left) and cross-linguistic colexification network (right). The figure illustrates how colexification networks can be reconstructed from cross-linguistic colexification data, using information obtained from the CLICS database (Version 3.0, Rzymski et al. 2020).

and (3) the general representation of data (§ 3.4). In the following, we will present previous studies devoted to cross-linguistic colexifications (§ 2.1), discuss the improvements in more detail (§ 3), illustrate their consequences for CLICS 4 (§ 4), and reflect on the future of cross-linguistic colexification data (§ 5).

2 Background

2.1 Cross-Linguistic Colexifications

Not long after François (2008) had first introduced the term *colexification* along with initial ideas on how the phenomenon could be analyzed using cross-linguistic data, typologists quickly adopted the term and the technique to study lexical semantics both globally and in certain linguistic areas. Two major reasons contributed to the popularity of the term and the technique.

First, polysemy and homophony are notoriously difficult to distinguish, specifically when analyzing languages whose history is less well known. While scholars sometimes distinguish both relations by degree of semantic similarity, arguing that homophonous words show greater divergence in meaning than polysemous words (Leivada and Murphy, 2021, 7), the original distinction between polysemy and homophony is strictly diachronic. Thus, they reflect two distinct processes of language change: polysemy is the result of semantic change, while homophony is the result of a merger of originally distinct word forms due to sound change (Sperber 1923, 12f, Apresjan 1974, 11). However, in the minds of speakers, the history of the words does not play a major role. Speakers seem to show some general awareness that some words have multiple senses that are closely related to each other, whereas other words with distinct senses merely sound alike (Enfield and Comrie, 2015, 20f). While it may seem useful to distinguish polysemy and homophony in theory, the distinction of the two relations in practice is difficult to make. Omitting the explicit distinction between the two forms of lexical ambiguity allowed scholars to assemble data in an unbiased and efficient way. Scholars could accumulate colexification data for their areas of interest without having to discuss the consequences of impractical terminology. Instead of deciding whether the findings would reflect polysemy or homophony, scholars could let the data decide, given that polysemy often largely exceeds homophony.

Second, scholars began to explore the benefits of modeling cross-linguistic colexifications with the help of network approaches (Cysouw, 2010). This not only led to clear visualizations of semantic similarities that could be observed across languages, but also opened up new possibilities for the analysis of cross-linguistic polysemy using network approaches (List et al., 2013) and the introduction of interactive techniques for data visualization and exploration, which later became a core component of the CLICS database (Mayer et al., 2014). Figure 1 illustrates how colexification networks can be constructed from colexification data, using data from CLICS 3 (Rzymski et al., 2020).

Due to this approach, which facilitates the collection of data and offers new ways to analyze the data through inspection and computation, cross-linguistic colexifications have become an integral part of lexical typology, with a multitude of applications in studies on semantic similarity. CLICS offered the first and largest collection of cross-linguistic colexifications and was used in several studies, examining a large number of topics, ranging from investigations on genealogical language relations (Blevins and Sproat, 2021; Blum et al., 2024) and linguistic areas (Gast and Koptjevskaja-Tamm, 2019), via analyses of particular semantic domains (Jackson et al., 2019; Di Natale et al., 2021; Brochhagen and Boleda, 2022; Tjuka et al., 2024), up to initial applications in computational linguistics (Bao et al., 2021, 2022) and communi-

cation science (Bradford et al., 2022). In addition, CLICS is now regularly consulted in typological studies that explore particular phenomena in detail, allowing authors to contrast their findings with their insights on a specific group of languages (Sjöberg, 2023; Souag, 2022; Schapper, 2019, 2022).

To summarize, cross-linguistic colexifications and cross-linguistic colexification networks have become a crucial tool in comparative linguistics. The application of cross-linguistic colexification analysis is not restricted to lexical typology, but provides interesting insights into additional fields of linguistics and beyond, including historical linguistics, areal linguistics, computational semantics, and human cognition.

2.2 Data Aggregation and Analysis in CLICS

The integral part of the Database of Cross-Linguistic Colexifications is the workflow by which data are aggregated from individual datasets and later analyzed to create a colexification network. To be able to aggregate data from different resources, datasets must be standardized. Standardization is achieved with the help of Cross-Linguistic Data Formats (CLDF, https://cldf.clld.org, Forkel et al., 2018), an initiative that builds on the CSVW standard for tabular data on the web (https://csvw.org, Gower, 2021), but extends CSVW with semantics relevant to comparative linguistics. A CLDF dataset is a collection of CSV files linked via a JSON file that stores the metadata, providing information on how the CSV files should be interpreted computationally and what values are shared across the files. Thus, a CLDF dataset is a small relational database with specific semantics that link the data with additional data from outside.

The most important external datasets that CLICS links to are three *reference catalogs*: *Glottolog*, *Concepticon*, and *CLTS*. Glottolog (https://glottolog.org, Hammarström et al., 2025) offers basic information on language varieties, including information on language classification, geolocations, and the documentation status of individual languages. Concepticon (https://concepticon.clld.org, List et al., 2025) offers a collection of basic senses that are expressed across multilingual word lists. Senses are provided in the form of *concept sets* that are linked across several hundred *concept lists* that have been annotated by the Concepticon team in the past decade (for details on the curation process, see Tjuka et al., 2023). CLTS (https://clts.clld.org, List et al., 2021) is a reference catalog for *Cross-Linguistic Transcription Systems* that standardizes phonetic transcriptions by advocating a subset of the International Phonetic Alphabet (IPA, International Phonetic Association, 1999) that is represented in the form of distinctive features (for details, see Anderson et al. 2018 and Rubehn et al. 2024). The conversion of individual datasets to the CLDF standard is supported by dedicated Python libraries (most importantly the `CLDFBench` packages, Forkel and List 2020) that help to check the overall consistency of the data.

From a collection of CLDF datasets, the CLICS aggregation workflow iterates over the datasets and assembles cross-linguistic colexifications for each language variety. Here, CLICS uses an efficient method that avoids comparing n words in one language against n words in the same language, but rather identifies colexifications from tuples, consisting of a word form and its corresponding sense (a *concept set* in the Concepticon catalog), with the help of hash tables (List, 2022). In other words, the method iterates over all words in a dataset only once, instead of comparing all words against each other, which would result in large computation times.

Having created a large colexification network of all CLDF datasets, the CLICS workflow analyzes this data further by computing *communities*, that is, partitions of nodes in a graph that show more connections to each other than to other nodes outside of the partition (Newman, 2006, 8577). Communities are inferred with the help of the Infomap algorithm (Rosvall and Bergstrom, 2008) and are used to structure the web application, by allowing users to inspect either entire communities or individual subsets of the data. The methods for data aggregation and analysis are freely accessible and can be easily applied by scholars to create their analyses of subsets of the data in CLICS or by extending the CLICS collection further, as illustrated, for example, by Tjuka (2024b).

2.3 Shortcomings of CLICS

Although the CLICS database serves as a main provider of cross-linguistic information on colexifications, CLICS 3 showed four major shortcomings that need to be addressed to ensure that future findings based on the data are solid and reliable.

The first shortcoming relates to the data underlying CLICS. While data from 30 datasets were aggregated in Version 3.0 (Rzymski et al., 2020),

many more datasets have recently been made available via the Lexibank repository (List et al., 2022). Improving the database by increasing the number of datasets is thus one of the most urgent tasks that should be addressed in an updated version.

The second shortcoming relates to the treatment of concepts in the database. CLICS 3 used a rather naïve approach by taking concept sets provided by the Concepticon reference catalog at face value, without considering their interdependencies. Concepticon has several concept sets that appear in a hierarchical relation to other concept sets, mostly reflecting cases of underspecification, such as the concept set BLUE OR GREEN, expressed in the Vietnamese word *xanh*. The colexification inference workflow in CLICS 3 treats the colexification of BLUE and GREEN expressed by the word *xanh* as a single concept. However, this omits valuable colexification information.

Third, CLICS 3 provided information from more than 3,000 language varieties. However, a closer look at the data showed that only a small proportion of the included languages met the requirement set by the editors of CLICS 3 to provide elicitation glosses for at least 250 concepts. For CLICS 3, the authors instead selected 30 datasets that were officially compiled from concept lists with 250 or more items. The resulting word lists for individual languages, however, were often scarce and a larger number of the languages did not meet the originally stated coverage criterion.

Fourth, CLICS 3 offered the colexification network exclusively in the form of a GML file. Although GML is a common format for the encoding of graphs (Himsolt, 2010), accepted by many software tools, including `igraph` (https://igraph.org, Csárdi and Nepusz, 2006), `NetworkX` (https://networkx.org/, Hagberg et al., 2008), and `Cytoscape` (http://cytoscape.org/, Smoot et al., 2011), the format is not well-suited to share the extensive data on colexification patterns computed by CLICS 3. As a result, more transparent data formats for handling colexification data and colexification networks are needed to represent the results of the CLICS workflow in detail.

With the increasing use of CLICS 3, it is time to tackle these four points of criticism. In this study, we address these shortcomings by creating an updated version of CLICS that substantially increases the amount of data, improves the handling of concepts, corrects for the bias in language and concept selection, and makes the data representation more transparent.

3 Materials and Methods

In the following, we will introduce all necessary steps that lead to the creation of our modified CLICS 4 database. We followed the established workflow for data aggregation used in CLICS 3 to some extent (§ 2.2). However, we present a drastic increase of data based on standardized datasets (§ 3.1), introduce an improved handling of concepts during data aggregation (§ 3.2), refine the selection of languages and concepts (§ 3.3), and make the representation of the colexification data more transparent (§ 3.4).

3.1 Data Basis

CLICS 3 was based on 30 datasets available in CLDF. Many more datasets have since been published as part of the Lexibank repository, which was first published in 2022 (List et al., 2022) as Lexibank 1 and curates data from 100 different datasets of different sizes. Of those 52 Lexibank datasets were suitable for inclusion in CLICS, because they were based on concept lists that contain 250 or more items (this criterion was used to build CLICS 3, Rzymski et al. 2020). The newest version, Lexibank 2, offers data for 134 different datasets that are all phonetically transcribed (Blum et al., 2025). For our enhanced version of CLICS, we identified 95 suitable datasets. These datasets are listed in the supplementary material accompanying this study.

The datasets include cross-linguistics studies of specific language groups (e.g., Bowern and Atkinson, 2012; Bodt and List, 2019) and global collections such as the Intercontinental Dictionary Series (IDS, https://ids.clld.org, Key and Comrie, 2023) or the World Loanword Database (https://wold.clld.org, Haspelmath and Tadmor, 2009). The latter datasets were not originally provided together with phonetic transcriptions, but recent studies have added them (see Miller et al. 2020 for WOLD and List 2023 and Miller and List 2024 for IDS).

3.2 Concept Handling

The colexifications in CLICS result from comparing words mapped to the standardized concept sets in Concepticon (List et al., 2016; Tjuka et al., 2023). The consequent mapping of the elicitation

glosses in individual datasets to the Concepticon reference catalog has been one of the most important factors that allowed for the growth of CLICS: Version 1.0 (List et al., 2014) containing 221 language varieties and 1,280 concepts, Version 2.0 (List et al., 2018) containing 1,220 language varieties and 2,487 concepts, and Version 3.0 (Rzymski et al., 2020) containing 3,156 language varieties and 2,906 concepts. However, through the mapping of the datasets to the Concepticon, a bias for a certain number of concepts that exhibit hierarchical relations to other concepts was introduced.

Already with its first launch (List et al., 2016), Conception has allowed for the definition of broad concepts that are expressed as such only in specific languages or specific linguistic areas. As an example, consider the concept sets ARM OR HAND and FOOT OR LEG. These concept sets are expressed by individual word forms in languages such as Vietnamese *tay*, referring to 'arm' or 'hand', or Russian *noga*, referring to 'foot' or 'leg'. However, many languages distinguish them further, using individual words for ARM, HAND, FOOT, and LEG, respectively.

Some lists in Concepticon have a linguistic area or language family as a target. Thus, the introduction of underspecified concept sets, such as ARM OR HAND or FOOT OR LEG was important, because linguists reporting on Slavic languages or particular languages in South-East Asia do not elicit both ARM and HAND, if they know that these are always colexified in the languages under study. However, this kind of lexical underspecification, as we encounter it in the lexicons of Vietnamese and Russian, is one of the typical reasons for colexifications. Therefore, it is important to list such cases as true colexifications of ARM and HAND, as well as FOOT and LEG. The original aggregation technique used by CLICS ignores these cases. As a result, important colexification information for a large number of languages is lost.

In our updated version CLICS 4, we account for underspecification directly, by defining a list of 85 concept sets that exhibit underspecification along with the more specific target concepts that they cover. While most of these underspecified concept sets can be represented by two concept sets, some are represented by more than two (specifically kinship terms like SISTER, which has four counterparts: YOUNGER SISTER (OF MAN), YOUNGER SISTER (OF WOMAN), OLDER SISTER (OF MAN), and OLDER SISTER (OF WOMAN)). In addition, we decided to replace some concept sets with a too broad or too narrow definition by more common concept sets (e.g. replacing STONE OR ROCK by STONE because ROCK did not occur in the data).

When encountering words that are mapped to these concepts during the initial iteration over all word lists in the data, the respective words are multiplied and each of the words is mapped to the specific concept sets covered by the underspecified concept sets. Word forms that are artificially multiplied in this form are marked in the resulting dataset by providing information on the original concept set. In total, we identify 85 underspecified concept sets in Concepticon that are relevant for the data in our modified version of CLICS. Of the 1,445,845 word forms in CLICS 4, 107,921 word forms result from this refinement procedure. A detailed list of the concept replacements can be found in Appendix A.

3.3 Language and Concept Selection

CLICS 3 included data from 3,156 language varieties. The criterion for including a given word list in the database was the size of the concept list underlying the respective dataset. The idea was to include only those languages with word forms for 250 or more concepts. However, since the editors of CLICS 3 only checked the size of the concept lists at the level of entire datasets, the CLICS 3 data contained a large number of language varieties with much fewer than 250 concepts covered. When discarding those varieties that contain fewer than 250 word forms, only 1,674 varieties remain.

After detecting this problem when reviewing individual datasets in CLICS 3, we decided to modify the criterion for the selection of languages in three ways. First, instead of setting the threshold to 250 words per language, we lowered it to 180 words, accounting for the fact that almost half of the languages in CLICS 3 would not pass this threshold. The threshold was chosen because we noticed that there were many datasets with 200 words or fewer. For many languages, only versions of the Swadesh list with 200 concepts (Swadesh, 1952) are available, so the chance of obtaining some concepts missing for individual languages is considerably high. Setting the threshold a bit lower allows us to predefine a core set of concepts that are comparable across languages (and which could be modified anytime, depending on the analysis one desires to conduct). Second, in our modified data aggregation workflow, the threshold is applied to individual lan-

guage varieties rather than to entire datasets. This means that for all languages in the sample, we count whether they meet the inclusion criterion or not. As a result, it may happen that only certain parts of the datasets from which CLICS 4 aggregates the word lists make it into the final database. Third, in order to yield a more meaningful selection of concepts, our workflow first orders all concepts by their occurrence across the languages in the data and then retains the most frequent 1,800 concepts. When aggregating the data from the individual word lists, only these concepts are retained. This procedure helps to decrease the sparsity of the data, resulting from the fact that the individual word lists often differ quite drastically with respect to the concepts for which they provide elicitation glosses. While the cutoff point may seem arbitrary, it reflects our experience in working with the mapping of concept lists in the Concepticon project: beyond 1,800 concepts, the chances of finding concepts expressed across many languages from many different families drop considerably.

3.4 Data Representation

The CLICS 3 colexification data was shared in the form of an SQLite database, while the network information was shared in the form of a GML file, offering the colexification networks with nodes, edges, and specific node and edge attributes. It was not a difficult task to implement the CLICS 3 workflow because the GML format can be easily read by different software packages. However, working with the data revealed several shortcomings of the GML format as the exclusive format for sharing the colexification network.

When following the core principle of CLDF in using tables as the basic representation format wherever possible, it would be straightforward to represent a graph with the help of two tables. One table would represent the nodes of a graph, with node attributes being provided in additional columns, and another table would represent the edges, with edge attributes being represented in additional columns. It turned out that this format could not only be easily represented in the CLDF specification, but that it would allow us to represent colexification data in the form of a *structural dataset* (Forkel et al., 2018). While the primary dataset underlying CLICS 4 provides information on colexifications between a fixed set of standardized concept sets, the additional view as a structural dataset – resembling a cross-linguistic typological database – offers a language-centered view: colexifications are modeled as parameters and for each language we provide information on their presence or absence. Thus, following (Forkel and List, 2020) in combining a word list and a structural dataset in a unified CLDF dataset, CLICS 4 now consists of a large aggregated word list with individual word forms across several thousand language varieties, along with structural data that provides information on the languages that exhibit certain colexifications.

Structural data in CLDF typically consist of a *parameter table* that provides information on the features comparable across languages, and a *value table* that provides information on the individual values as they are reflected in individual languages. In our new data model for cross-linguistic colexification data, all individual colexifications that can be inferred when analyzing the aggregated word list feature are represented as *parameters*. In contrast, the corresponding values for each language are represented by three different codes, indicating if the feature represented by the parameter is *present*, *absent*, or *missing*. Thus, our proposal for CLICS 4 not only informs whether a given language exhibits a particular colexification but also whether it does *not* show the colexification, or whether the information is missing, since elicitation glosses for at least one of the concepts involved in the colexification are missing in the word list.

There are two major advantages of this new representation. The first advantage is that colexifications can be directly inspected in tabular form. Since the colexification data are shared in a table format as part of the CLDF dataset underlying CLICS 4, interested users can browse through the colexifications using their favorite spreadsheet editor. Analyzing the colexification network with software tools is also facilitated, given that all major tools support tabular data. This means that networks can be conveniently analyzed computationally or visualized with graph visualization software, such as Cytoscape (for a tutorial, see Tjuka, 2024a). The second advantage is that it is much easier to integrate the data produced by CLICS 4 with the data shared by other projects. Community assignments, along with additional information on the coverage of concepts across languages and language families, for example, are now part of the concept table that serves as the basic parameter table for the CLICS 4 word list. From this representation, it is easy to integrate the data not only into the

Concepticon (see also Bocklage et al., 2024) but also into extended reference catalogs such as NoRaRe (https://norare.clld.org, Tjuka et al., 2022), a catalog that extends the Concepticon by providing additional information on *norms*, *rates*, and *ratings* for words and concepts across multiple languages.

3.5 Implementation

CLICS 4 is implemented in the form of a `CLDFBench` package (Forkel and List, 2020), written in Python, that can be installed from the command line and contains the resulting CLDF data along with the code that was used to create the data. The package is shared as part of the supplemental material accompanying this study and contains additional information and code examples that were used to produce the findings presented in this study.

4 Data Validation

4.1 Comparing CLICS 3 and CLICS 4

In order to understand the differences between our updated version CLICS 4 and the previous versions of CLICS, most importantly the last officially published version CLICS 3 by Rzymski et al. (2020), we carried out a detailed comparison of CLICS 3 and CLICS 4. Given that we deliberately restricted the number of concepts in CLICS 4 to an initial list of 1,800 concepts – of which 1,730 were retained when selecting those languages that would cover at least 180 concepts of the initial list – it may seem as if CLICS 4 simply reduced the amount of data in contrast to CLICS 3. However, this is not the case, which is apparent when comparing the number of words, language varieties, languages (different glottocodes), and language families covered in both datasets, as shown in Table 1. CLICS 4 exceeds CLICS 3 not only regarding the number of language families and language varieties covered, but most notably with respect to the number of word forms that are provided in phonetic transcriptions. CLICS 4 reaches almost the same size as CLICS 3, while providing almost three times as many phonetic transcriptions.

A similar situation arises when comparing the overall number of concepts with the average number of languages and families *expressing* a concept in both datasets (also shown in Table 1). Here, CLICS 3 exceeds CLICS 4 in the number of concepts that are colexified (1,386 vs. 1,647), while showing similar values for the average number of languages expressing a concept (607 vs. 624).

Criterion	CLICS 3	CLICS 4
Datasets	30	95
Varieties	3 156	3 432
Languages	2 280	2 152
Families	200	247
Words	1 462 125	1 445 845
Transcriptions	563 878	1 445 845
Words per Variety	467	421
Concepts	2 906	1 730
Colexified Concepts	1 647	1 386
Languages per Concept	624	607
Families per Concept	61	92
Colexifications	4 228	3 986
Average Degree	5	6
Average Weighted Degree	36	53
Communities	249	315
Concepts per Community	6.6	4.4

Table 1: Comparison between CLICS 3 and CLICS 4. Colexifications are only counted when occurring in at least three different language families. Weighted degree is calculated by counting the number of language families per link.

However, regarding the average number of families expressing a concept, CLICS 4 largely exceeds CLICS 3 (92 vs. 61).

In sum, the comparison provided in Table 1 shows that CLICS 4 does not simply provide *more* data, resulting in more languages, more concepts, and more colexifications. Instead, the major improvements concerning the data basis, concept handling, and language selection yield a colexification network that consolidates the tendencies in the data rather than diversifying them further. Thus, while CLICS 4 has fewer colexified concepts, i.e., concepts that are part of a colexification, the concepts in the colexification network of CLICS 4 have more connections across more language families on average, as reflected in their degree distribution (6 vs. 5). In addition, these connections are also substantiated by more colexifications, as reflected in the weighted degree distribution (53 vs. 36). This trend can also be observed when directly comparing the inferred colexifications. There are 2,874 colexifications observed in both networks, 1,354 unique to CLICS 3, and 1,112 unique to CLICS 4. Of the common edges, 859 colexifications in CLICS 4 can be found in more language families, compared to 778 colexifications in CLICS 3.

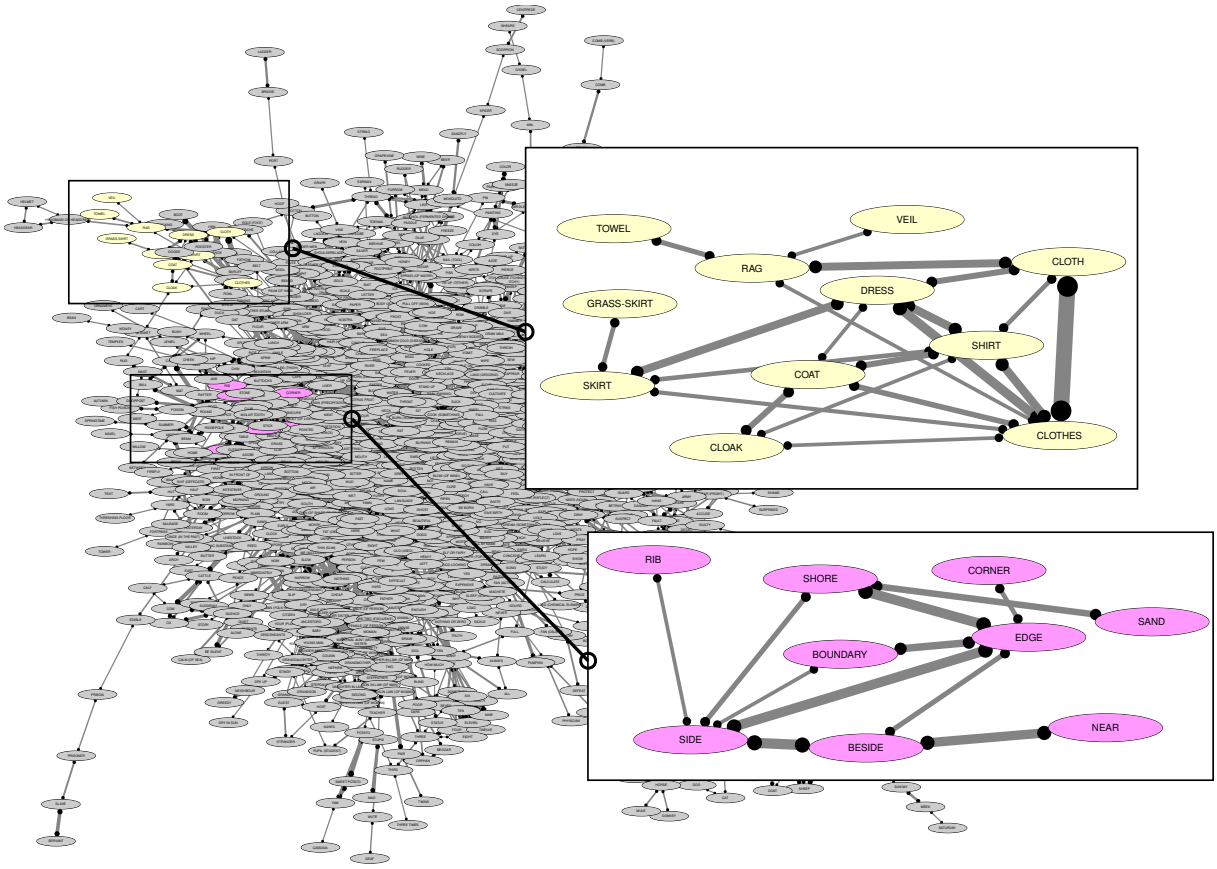


Figure 2: CLICS 4 colexification network with two selected communities (central concepts DRESS and EDGE).

4.2 Visualizing the CLICS 4 Network

To create a visual representation of the CLICS 4 network, researchers can either use the GML file that is provided along with the CLDF data, or the table with all colexifications that is shared as part of CLDF directly. As mentioned in § 3.4, the new data representation in tabular form as part of a unified CLDF dataset makes it easier to analyze the data computationally. The visualization of the data is also greatly facilitated, given that edge tables are the basic input format for network visualization software tools like Cytoscape. A tutorial on how to create a network visualization with Cytoscape is provided in Tjuka 2024a. We used this approach to create Figure 2, which provides a bird's eye view of the CLICS 4 network.

The figure shows the entire network with two communities highlighted and enlarged. The first community has the concept DRESS as a central node and shows colexifications with other clothing items. The edge weights represent the frequency with which a given colexification occurs across languages. For example, the colexification between DRESS and SKIRT is more frequent than the colexification with COAT. The second community has the concept EDGE as a central node and includes cross-linguistically frequent colexifications such as EDGE and SIDE and less frequent ones like EDGE and CORNER. Given the straightforward representation of the colexification network in CLICS 4, the data can conveniently be explored. By using Cytoscape, researchers can further investigate the properties of the network and filter them according to their particular research interests.

5 Conclusion and Outlook

We presented CLICS 4, an enhanced version of the Cross-Linguistic Colexification Database, which integrates lexical data for 3,432 language varieties, corresponding to 2,152 distinct Glottocodes. When creating CLICS 4 we used an advanced workflow for the aggregation and analysis of cross-linguistic colexification data that is based on an increased and improved data basis, an improved handling of concepts, more fine-grained criteria for the selection of languages and concepts, and an updated representation of the colexification data.

In contrast to previous colexification databases, CLICS 4 determines colexifications exclusively

based on phonetic transcriptions. This makes the data more consistent and robust and opens new possibilities to analyze the data in comparative studies. Due to the phonetic transcriptions in CLICS 4, future studies can build on the initial work to infer and investigate partial colexifications (List, 2023; Tjuka and List, 2024; Rubehn and List, 2025). In addition, phonetic transcriptions enable scholars to carry out more fine-grained analyses of colexifications inside specific language families, where a handling of cognate words is important to identify colexifications that have evolved independently from colexifications that have been inherited across branches (Tjuka et al., 2024).

Future studies can use CLICS 4 to explore the relationship between words and their meanings across a wide range of languages and uncover important insights into language evolution, cultural variations, and cognitive principles. In this way, CLICS 4 has great potential to contribute to future studies that address open questions in a broad range of linguistic subfields, including linguistic typology, historical linguistics, psycholinguistics, and computational linguistics.

Supplementary Material

All data and code underlying this study, along with instructions on how to run the code, are openly available. The CLICS 4 database is curated on GitHub (https://github.com/clics/clics4/tree/v0.5, Version 0.5) and archived with Zenodo (DOI: https://doi.org/10.5281/zenodo.16900180).
The code that we used to compare CLICS 3 and CLICS 4 is cuarated on Codeberg (https://codeberg.org/calc/clics4-paper/src/tag/v1.0, Version 1.0) and archived with Zenodo (DOI: https://doi.org/10.5281/zenodo.16902185).

Limitations

General limitations that apply to large-scale aggregation studies in comparative linguistics also apply to CLICS 4. These include the fact that the word list approach for aggregating cross-linguistic colexifications may fail to model fine-grained aspects of colexifications in individual language families, many of which cannot be modeled appropriately without a detailed inspection of particular languages and their history. An additional problem of all cross-linguistic colexification databases is that they contain a lot of missing data, showing low coverage for most concepts cross-linguistically. We also emphasize that detailed studies investigating the properties of CLICS 4 are missing so far, but we envisage that these will be carried out by different teams (not only including the team which compiled the data by now). Another improvement that needs to be implemented in the future is the treatment of some artificially separated concepts. For example, the current version splits the concept THINK into the more specific concepts THINK (REFLECT) and THINK (BELIEVE). While this modification reflects the ambiguity of the concept THINK, we suspect that there is no frequently used questionnaire for cross-linguistic data that would contain both THINK (REFLECT) and THINK (BELIEVE). As a result, one may call the colexification between THINK (REFLECT) and THINK (BELIEVE) in question, given that the database lacks direct evidence. This holds to an even larger degree for kinship terms. One solution we could think of would be to consider only THINK, as the broadest concept, because this concept is present in most languages. While our current technology would allow for such a handling, we think addressing this problem in a principled way will require a more thorough revision, potentially accompanied by additional computational analyses and very detailed decisions that should not be made in an ad-hoc style.

So far, CLICS 4 is limited to the data and the database itself can only be investigated with tools for network visualization and with computational approaches. As of now, the web application at https://clics.clld.org still serves the data underlying CLICS 3. Implementing the web application for CLICS 4 is planned and will follow in the near future.

Acknowledgments

This project was supported by the ERC Consolidator Grant ProduSemy (JML; Grant No. 101044282, see https://doi.org/10.3030/101044282) and the ERC Synergy Grant QUANTA (CR; Grant No. 951388, see https://doi.org/10.3030/951388). Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency (nor any other funding agencies involved). Neither the European Union nor the granting authority can be held responsible for them.

References

Cormac Anderson, Tiago Tresoldi, Thiago Costa Chacon, Anne-Maria Fehn, Mary Walworth, Robert Forkel, and Johann-Mattis List. 2018. A cross-linguistic database of phonetic transcription systems. *Yearbook of the Poznań Linguistic Meeting*, 4(1):21–53.

Juri D. Apresjan. 1974. Regular polysemy. *Linguistics*, 12(142):5–32.

Hongchang Bao, Bradley Hauer, and Grzegorz Kondrak. 2021. On universal colexifications. In *Proceedings of the 11th Global WordNet Conference*, pages 1–7, University of South Africa. Global WordNet Association.

Hongchang Bao, Bradley Hauer, and Grzegorz Kondrak. 2022. Lexical resource mapping via translations. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7147–7154, Marseille, France. European Language Resources Association.

Juliette Blevins and Richard Sproat. 2021. Statistical evidence for the Proto-Indo-European-Euskarian Hypothesis: A word-list approach integrating phonotactics. *Diachronica*, 0(0):1–59.

Frederic Blum, Carlos Barrientos, Johannes Englisch, Robert Forkel, Simon J. Greenhill, Christoph Rzymski, and Johann-Mattis List. 2025. Lexibank 2: Precomputed features for large-scale lexical data [version 2; peer review: 3 approved]. *Open Research Europe*, 5(126):1–24.

Frederic Blum, Carlos Barrientos, Adriano Ingunza, and Johann-Mattis List. 2024. Cognate reflex prediction as hypothesis test for a genealogical relation between the Panoan and Takanan language families. *Scientific Reports*, 14(30636):1–12.

Katja Bocklage, Anna Di Natale, Annika Tjuka, and Johann-Mattis List. 2024. *Directional tendencies in semantic change*. Humanities Commons.

Timotheus A. Bodt and Johann-Mattis List. 2019. Testing the predictive strength of the comparative method: An ongoing experiment on unattested words in Western Kho-Bwa languages. *Papers in Historical Phonology*, 4:22–44.

Claire Bowern and Quentin Atkinson. 2012. Computational phylogenetics and the internal structure of Pama-Nyungan. *Language*, 88(4):817–845.

Laurestine Bradford, Guillaume Thomas, and Yang Xu. 2022. Communicative need modulates lexical precision across semantic domains: A domain-level account of efficient communication. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, pages 2561–2568.

Thomas Brochhagen and Gemma Boleda. 2022. When do languages use the same word for different meanings? The Goldilocks Principle in colexification. *Cognition*, 226:1–8.

Gábor Csárdi and Tamás Nepusz. 2006. The Igraph software package for complex network research. *InterJournal Complex Systems*, 1695.

Michael Cysouw. 2010. Drawing networks from recurrent polysemies. *Linguistic Discovery*, 8(1):281–285.

Anna Di Natale, Max Pellert, and David Garcia. 2021. Colexification networks encode affective meaning. *Affective Science*, 2:99–111.

Nick J. Enfield and Bernard Comrie, editors. 2015. *Languages of Mainland South-East Asia. The state of the art*. Mouton de Gruyter, Berlin and New York.

Robert Forkel and Johann-Mattis List. 2020. CLDF-Bench: Give your cross-linguistic data a lift. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6995–7002, Marseille, France. European Language Resources Association.

Robert Forkel, Johann-Mattis List, Simon J. Greenhill, Christoph Rzymski, Sebastian Bank, Michael Cysouw, Harald Hammarström, Martin Haspelmath, Gereon A. Kaiping, and Russell D. Gray. 2018. Cross-Linguistic Data Formats, advancing data sharing and re-use in comparative linguistics. *Scientific Data*, 5(1):1–10.

Alexandre François. 2008. Semantic maps and the typology of colexification: Intertwining polysemous networks across languages. In Martine Vanhove, editor, *From polysemy to semantic change: Towards a typology of lexical semantic associations*, pages 163–215. John Benjamins, Amsterdam.

Volker Gast and Maria Koptjevskaja-Tamm. 2019. The areal factor in lexical typology: Some evidence from lexical databases. In Daniël Van Olmen, Tanja Mortelmans, and Frank Brisard, editors, *Aspects of Linguistic Variation*, pages 43–82. Walter de Gruyter, Berlin.

Robin Gower. 2021. *CSV on the Web*. Swirrl, Stirling.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference*, pages 11–15, Pasadena.

Harald Hammarström, Martin Haspelmath, Robert Forkel, and Sebastian Bank. 2025. *Glottolog [Dataset, Version 5.2.1]*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Martin Haspelmath and Uri Tadmor. 2009. The Loanword Typology Project and the World Loanword Database. In Martin Haspelmath and Uri Tadmor, editors, *Loanwords in the World's Languages*, pages 1–34. De Gruyter Mouton, Berlin.

Michael Himsolt. 2010. GML: A portable graph file format. Technical report, Universität Passau.

IPA, International Phonetic Association. 1999. *Handbook of the International Phonetic Association*. Cambridge University Press, Cambridge.

Joshua Conrad Jackson, Joseph Watts, Teague R. Henry, Johann-Mattis List, Robert Forkel, Peter J. Mucha, Simon J. Greenhill, Russell D. Gray, and Kristen A. Lindquist. 2019. Emotion semantics show both cultural variation and universal structure. *Science*, 366:1517–1522.

Mary Ritchie Key and Bernard Comrie. 2023. *The Intercontinental Dictionary Series [Dataset, Version 4.3]*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Evelina Leivada and Elliot Murphy. 2021. Mind the (terminological) gap: 10 misused, ambiguous, or polysemous terms in linguistics. *Ampersand*, 8:1–9.

Johann-Mattis List. 2022. How to compute colexifications with CL Toolkit (How to do X in linguistics 10). *Computer-Assisted Language Comparison in Practice*, 5(6).

Johann-Mattis List. 2023. Inference of partial colexifications from multilingual wordlists. *Frontiers in Psychology*, 14:1–10.

Johann-Mattis List, Cormac Anderson, Tiago Tresoldi, and Robert Forkel. 2021. *Cross-Linguistic Transcription Systems [Dataset, Version 2.3.0]*. Max Planck Institute for the Science of Human History, Jena.

Johann-Mattis List, Michael Cysouw, and Robert Forkel. 2016. Concepticon: A resource for the linking of concept lists. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, pages 2393–2400, Portorož, Slovenia. European Language Resources Association.

Johann-Mattis List, Robert Forkel, Simon J. Greenhill, Christoph Rzymski, Johannes Englisch, and Russell D. Gray. 2022. Lexibank, a public repository of standardized wordlists with computed phonological and lexical features. *Scientific Data*, 9(1):1–16.

Johann-Mattis List, Simon J. Greenhill, Cormac Anderson, Thomas Mayer, Tiago Tresoldi, and Robert Forkel. 2018. CLICS^2: An improved database of cross-linguistic colexifications assembling lexical data with the help of Cross-Linguistic Data Formats. *Linguistic Typology*, 22(2):277–306.

Johann-Mattis List, Thomas Mayer, Anselm Terhalle, and Matthias Urban. 2014. *CLICS: Database of Cross-Linguistic Colexifications [Dataset, Version 1.0]*. Forschungszentrum Deutscher Sprachatlas, Marburg.

Johann-Mattis List, Anselm Terhalle, and Matthias Urban. 2013. Using network approaches to enhance the analysis of cross-linguistic polysemies. In *Proceedings of the 10th International Conference on Computational Semantics*, pages 347–353, Potsdam, Germany. Association for Computational Linguistics.

Johann-Mattis List, Annika Tjuka, Frederic Blum, Alžběta Kučerová, Carlos Barrientos Ugarte, Christoph Rzymski, Simon J. Greenhill, and Robert Forkel. 2025. *CLLD Concepticon [Dataset, Version 3.4.0]*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Thomas Mayer, Johann-Mattis List, Anselm Terhalle, and Matthias Urban. 2014. An interactive visualization of crosslinguistic colexification patterns. In *Proceedings of the LREC Workshop 'VisLR: Visualization as Added Value in the Development, Use and Evaluation of Language Resources'*, pages 1–8, Reykjavik, Iceland. European Language Resources Association.

John Miller and Johann-Mattis List. 2024. Adding standardized transcriptions to Panoan and Tacanan languages in the Intercontinental Dictionary Series. *Computer-Assisted Language Comparison in Practice*, 7(2):69–77.

John E. Miller, Tiago Tresoldi, Roberto Zariquiey, César A. Beltrán Castañón, Natalia Morozova, and Johann-Mattis List. 2020. Using lexical language models to detect borrowings in monolingual wordlists. *PLOS One*, 15(12):e0242709.

M. E. J. Newman. 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Science of the United States of America*, 103(23):8577–8582.

Martin Rosvall and Carl T. Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123.

Arne Rubehn and Johann-Mattis List. 2025. Partial colexifications improve concept embeddings. In *Proceedings of the Association for Computational Linguistics 2025. Long Papers*, pages 20571–20586.

Arne Rubehn, Jessica Nieder, Robert Forkel, and Johann-Mattis List. 2024. Generating feature vectors from phonetic transcriptions in Cross-Linguistic Data Formats. *Proceedings of the Society for Computation in Linguistics*, 7(1):205–216.

Christoph Rzymski, Tiago Tresoldi, Simon J. Greenhill, Mei-Shin Wu, Nathanael Schweikhard, Maria Koptjevskaja-Tamm, Volker Gast, Timotheus A. Bodt, Abbie Hantgan, Gereon A. Kaiping, Sophie Chang, Yunfan Lai, Natalia Morozova, Heini Arjava, Nataliia Hübler, Ezequiel Koile, Steve Pepper, Mariann Proos, Briana Van Epps, Ingrid Blanco, Carolin Hundt, Sergei Monakhov, Kristina Pianykh, Sallona Ramesh, Russell D. Gray, Robert Forkel, and Johann-Mattis List. 2020. The Database of Cross-Linguistic Colexifications, reproducible analysis of cross-linguistic polysemies. *Scientific Data*, 7(1):1–12.

Antoinette Schapper. 2019. The ethno-linguistic relationship between smelling and kissing: A Southeast Asian case study. *Oceanic Linguistics*, 58(1):92–109.

Antoinette Schapper. 2022. Baring the bones: The lexico-semantic association of bone with strength in Melanesia and the study of colexification. *Linguistic Typology*, 26(2):313–347.

Anna Sjöberg. 2023. *Knowledge predication: A semantic typology*. Ph.D. thesis, Stockholm University, Stockholm.

Michael E. Smoot, Keiichiro Ono, Johannes Ruscheinski, Peng-Liang Wang, and Trey Ideker. 2011. Cytoscape 2.8: New features for data integration and network visualization. *Bioinformatics*, 27(3):431–432.

Lameen Souag. 2022. How a West African language becomes North African, and vice versa. *Linguistic Typology*, 26(2):283–312.

Hans Sperber. 1923. *Einführung in die Bedeutungslehre [Introduction to the study of meaning]*. Kurt Schroeder, Bonn and Leipzig.

Morris Swadesh. 1952. Lexico-statistic dating of prehistoric ethnic contacts. *Proceedings of the American Philosophical Society*, 96(4):452–463.

Annika Tjuka. 2024a. How to visualize colexification networks in Cytoscape (How to do X in linguistics 14). *Computer-Assisted Language Comparison in Practice*, 7(1):7–16.

Annika Tjuka. 2024b. Objects as human bodies: Cross-linguistic colexifications between words for body parts and objects. *Linguistic Typology*, pages 1–40.

Annika Tjuka, Robert Forkel, and Johann-Mattis List. 2022. Linking norms, ratings, and relations of words and concepts across multiple language varieties. *Behavior Research Methods*, 54(2):864–884.

Annika Tjuka, Robert Forkel, and Johann-Mattis List. 2023. Curating and extending data for language comparison in Concepticon and NoRaRe. *Open Research Europe*, 2(141):1–13.

Annika Tjuka, Robert Forkel, and Johann-Mattis List. 2024. Universal and cultural factors shape body part vocabularies. *Scientific Reports*, 14(1):1–12.

Annika Tjuka and Johann-Mattis List. 2024. Partial colexifications reveal directional tendencies in object naming. *Yearbook of the German Cognitive Linguistics Association*, 12(1):95–114.

A Original and Replaced Concepts

Original Concept	Rep. Con.	New Con.	Details
MOUNTAIN OR HILL	1466	2	HILL (733), MOUNTAIN (733)
SPRING OR WELL	1016	2	SPRING (OF WATER) (508), WELL (508)
STONE OR ROCK	484	1	STONE (484)
MAN	2280	1	MALE PERSON (2280)
BROTHER	2668	4	OLDER BROTHER (OF MAN) (667), OLDER BROTHER (OF WOMAN) (667), YOUNGER BROTHER (OF MAN) (667), YOUNGER BROTHER (OF WOMAN) (667)
OLDER BROTHER	2462	2	OLDER BROTHER (OF MAN) (1231), OLDER BROTHER (OF WOMAN) (1231)
YOUNGER BROTHER	1814	2	YOUNGER BROTHER (OF MAN) (907), YOUNGER BROTHER (OF WOMAN) (907)
SISTER	3060	4	OLDER SISTER (OF MAN) (765), OLDER SISTER (OF WOMAN) (765), YOUNGER SISTER (OF MAN) (765), YOUNGER SISTER (OF WOMAN) (765)
OLDER SISTER	1910	2	OLDER SISTER (OF MAN) (955), OLDER SISTER (OF WOMAN) (955)
YOUNGER SISTER	1664	2	YOUNGER SISTER (OF MAN) (832), YOUNGER SISTER (OF WOMAN) (832)
UNCLE	1254	2	MATERNAL UNCLE (MOTHER'S BROTHER) (627), PATERNAL UNCLE (FATHER'S BROTHER) (627)
AUNT	1406	2	MATERNAL AUNT (MOTHER'S SISTER) (703), PATERNAL AUNT (FATHER'S SISTER) (703)
HE OR SHE OR IT	3444	3	HE (1148), IT (1148), SHE (1148)
WE	3862	2	WE (EXCLUSIVE) (1931), WE (INCLUSIVE) (1931)
BLOOD VESSEL	342	1	VEIN (342)
ROAST OR FRY	868	2	FRY (434), ROAST (SOMETHING) (434)
SIEVE OR STRAIN	409	1	STRAIN (409)
TORCH OR LAMP	400	1	LAMP (400)
SICKLE OR SCYTHE	445	1	SICKLE (445)
BRANCH OR TWIG	353	1	BRANCH (353)
STRIKE OR BEAT	1416	2	BEAT (708), STRIKE (708)
CHOP	1116	2	CHOP (INTO PIECES) (558), CUT (WITH AXE) (558)
BREAK (DESTROY OR GET DESTROYED)	2240	2	BREAK (BREAKING) (1120), BREAK (CLEAVE) (1120)
TWIST (AROUND)	415	1	TWIST (415)
CRAWL OR CREEP	455	1	CRAWL (455)
STORE	311	1	SHOP (311)
AFTER	743	1	AFTERWARDS (743)
OLD	5164	2	OLD (AGED) (2582), OLD (USED) (2582)
BREATH OR BREATHE	728	2	BREATH (364), BREATHE (364)
BE ALIVE OR LIFE	990	2	BE ALIVE (495), LIFE (495)

BE DEAD OR DIE	1358	1	DIE (1358)
MIGHTY OR POWERFUL OR STRONG	852	2	POWERFUL (426), STRONG (426)
COOKING POT	660	1	POT (660)
DO OR MAKE	1582	2	DO (791), MAKE (791)
BRONZE OR COPPER	273	1	COPPER (273)
DOWN OR BELOW	646	2	BELOW OR UNDER (323), DOWN (323)
CENTER OR MIDDLE	337	1	MIDDLE (337)
BEGIN OR START	520	1	BEGIN (520)
CANNON OR GUN	338	1	GUN (338)
FINGERNAIL OR TOENAIL	872	2	FINGERNAIL (436), TOENAIL (436)
PATH OR ROAD	2920	2	PATH (1460), ROAD (1460)
COLD (OF WEATHER)	204	1	COLD (204)
A LITTLE	191	1	FEW (191)
HOW MANY	1592	2	HOW MANY PIECES (796), HOW MUCH (796)
SON-IN-LAW	434	2	SON-IN-LAW (OF MAN) (217), SON-IN-LAW (OF WOMAN) (217)
CUT (WITH KNIFE)	250	1	CUT (250)
MARRY (AS MAN)	269	1	MARRY (269)
HIT	2051	1	STRIKE (2051)
THIN (OF LEAF AND CLOTH)	240	1	THIN (OF SHAPE OF OBJECT) (240)
ITCH OR ITCHY OR ITCHING	344	2	ITCH (172), ITCH (CAUSE ITCHING OR FEEL ITCHY) (172)
HE OR SHE	2052	2	HE (1026), SHE (1026)
THIN	3456	2	THIN (OF SHAPE OF OBJECT) (1728), THIN (SLIM) (1728)
MALE	938	2	MALE (OF ANIMAL) (469), MALE (OF PERSON) (469)
FEMALE PERSON	1154	1	WOMAN (1154)
CHILD	3876	2	CHILD (DESCENDANT) (1938), CHILD (YOUNG HUMAN) (1938)
HIDE	2594	2	HIDE (CONCEAL) (1297), HIDE (ONESELF) (1297)
THINK	3834	2	THINK (BELIEVE) (1917), THINK (REFLECT) (1917)
SMELL	1608	2	SMELL (PERCEIVE) (804), SMELL (STINK) (804)
BOIL	338	1	BOIL (OF LIQUID) (338)
BURN	5012	2	BURN (SOMETHING) (2506), BURNING (2506)
KNOW	689	1	KNOW (SOMETHING) (689)
EAGLE OR HAWK	382	2	EAGLE (191), HAWK (191)
ARM OR HAND	720	2	ARM (360), HAND (360)
FOOT OR LEG	2340	2	FOOT (1170), LEG (1170)
FLESH OR MEAT	2852	2	FLESH (1426), MEAT (1426)

PERSPIRE OR SWEAT	996	2	SWEAT (PERSPIRE) (498), SWEAT (SUBSTANCE) (498)
THIN (OF HAIR AND THREAD)	34	1	THIN (OF SHAPE OF OBJECT) (34)
RAINING OR RAIN	1086	2	RAIN (PRECIPITATION) (543), RAIN (RAINING) (543)
BLACK OR DARK	204	2	BLACK (102), DARK (102)
EARTH OR LAND	402	2	EARTH (SOIL) (201), LAND (201)
TURN	2620	2	TURN (SOMETHING) (1310), TURN AROUND (1310)
BELLY OR STOMACH	70	2	BELLY (35), STOMACH (35)
FINGER OR TOE	4	2	FINGER (2), TOE (2)
WE TWO (INCLUSIVE)	302	1	WE TWO (302)
HOT OR WARM	274	2	HOT (137), WARM (137)
SHY OR ASHAMED	607	1	SHY (607)
NO OR NOT	2190	2	NO (1095), NOT (1095)
CLAW OR NAIL	759	3	CLAW (253), FINGERNAIL (253), TOENAIL (253)
BLUE OR GREEN	58	2	BLUE (29), GREEN (29)
BAD OR EVIL	1344	2	BAD (672), EVIL (672)
THATCH OR ROOF	1408	2	ROOF (704), THATCH (704)
PAINFUL OR SICK	1954	2	PAINFUL (977), SICK (977)
DREAMING OR DREAM	514	2	DREAM (257), DREAM (SOMETHING) (257)
LARGE WILD HERBIVORE	132	1	DEER (132)

FRIDA to the Rescue! Analyzing Synthetic Data Effectiveness in Object-Based Common Sense Reasoning for Disaster Response

Mollie Shichman[1], **Claire Bonial**[2], **Austin Blodgett**[2], **Taylor Pellegrin**[3], **Francis Ferraro**[4], **Rachel Rudinger**[1]
[1]University of Maryland, College Park, [2]Army Research Lab
[3]Oak Ridge Associated Universities, [4]University of Maryland, Baltimore County
mshich@umd.edu, claire.n.bonial.civ@army.mil,
ferraro@umbc.edu, rudinger@umd.edu

Abstract

During Human Robot Interactions in disaster relief scenarios, Large Language Models (LLMs) have the potential for substantial physical reasoning to assist in mission objectives. However, these reasoning capabilities are often found only in larger models, which are not currently reasonable to deploy on robotic systems due to size constraints. To meet our problem space requirements, we introduce a dataset and pipeline to create Field Reasoning and Instruction Decoding Agent (FRIDA) models. In our pipeline, domain experts and linguists combine their knowledge to make high-quality, few-shot prompts used to generate synthetic data for fine-tuning. We hand-curate datasets for this few-shot prompting and for evaluation to improve LLM reasoning on both general and disaster-specific objects. We concurrently run an ablation study to understand which kinds of synthetic data most affect performance. We fine-tune several small instruction-tuned models and find that ablated FRIDA models only trained on objects' physical state and function data outperformed both the FRIDA models trained on all synthetic data and the base models in our evaluation. We demonstrate that the FRIDA pipeline is capable of instilling physical common sense with minimal data.

1 Introduction

Which of the following would be most dangerous if it collapsed? This question, as seen in Figure 1, is fairly trivial for humans to answer, but requires several types of semantic knowledge. One must know the general size of these items and their other functions to fully assess the danger the item poses. A collapse is also a change of state that fundamentally shifts the use of these objects; a collapsed chair could be more likely to cut or scrape someone, but it could also mean the chair can now be carried if the chair folds. All of this knowledge is needed to answer this question, and all of it is embedded in our semantic understanding of objects that can cause danger and objects that can collapse, both intentionally and unintentionally.

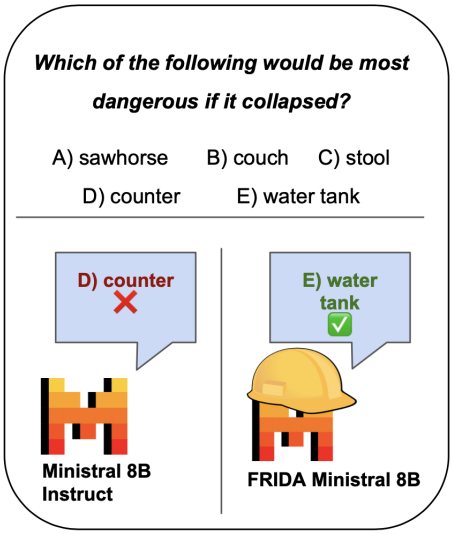


Figure 1: An example of how a FRIDA-tuned LLM outperforms its base model on questions combining an object's affordances and physical characteristics.

The ability to reason about objects is especially important in the context of human-robot interaction in disaster relief scenarios (Bonial et al., 2024). For example, during search and rescue after an earthquake, a robot needs to know how to navigate partially collapsed buildings and how to use the many tools required to free people from the rubble. However, using robots to aid in disaster relief introduces many constraints. Because of the destruction a disaster can wreak, consistent internet connectivity cannot be assumed. For human safety, robots must be handled via radio in a secure location. This low-bandwidth communication means limited image data can be transmitted to the handlers, which rules out remote piloting (Bonial et al., 2024). **We therefore need an autonomous system that can reason about its environment and the relief tasks**

Proceedings of the 16th International Conference on Computational Semantics, pages 16–29
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

required.

As LLMs have improved dramatically, their abilities at semantic reasoning about objects have improved as well. LLMs have long been proven able to encode physical world knowledge (Petroni et al., 2019), and their embeddings can improve physical understanding of an environment and its objects both within and beyond a fine-tuned domain (Cohen et al., 2024).

However, much of this improvement is found only in larger models trained on more data (Wei et al., 2022a; Kaplan et al., 2020). This makes these essential semantic capabilities less accessible to our use case. Our robot cannot rely on an internet connection to make API calls. We instead must utilize the robot's limited on-board computing power, which can be as little as 16 GB of virtual RAM on an array of GPUs (Osteen, 2025). That amount of GPU RAM can only reasonably run inference on a 13 Billion parameter model given the heuristics described in Anthony et al. (2023). Furthermore, this heuristic assumes that our robot is not running other processes in parallel, which is fairly unreasonable. We thus wanted to answer: **Given our constraints, how can we imbue all the physical common sense and semantics needed for smaller LLMs to be more capable at understanding a disaster environment?**

To answer our research question, we first tested the effectiveness of fine-tuning smaller models on disaster relief data. However, available data proved to be an additional constraint. Most publicly available data on disasters is social media-based reactions (Godinho, 2024), which do not pertain much to our subdomain of disaster relief efforts. Furthermore, the specific knowledge (and to a lesser extent, the general knowledge) required for each mission varies by disaster. For example, after an earthquake, a robot needs to find survivors, while after a chemical spill, a robot needs to sample the environment for hazardous materials. **Therefore, we need a method for generating training data for specific disasters, and we need to evaluate which data are most effective at improving robot performance.**

We present a pipeline to create Field Reasoning and Instruction Decoding Agent (FRIDA) models as a proof of concept for LLM viability in the disaster relief domain. For FRIDA, we leveraged both disaster and linguistic expertise to create gold-standard instructions that, in turn, are used as a basis for synthetic data generation, as seen in Figure 2. These synthetic data are then used to fine-tune smaller models that fit our memory constraints. Like its rescue dog eponym,[1] our FRIDA models were initially developed and tested for earthquake disaster relief, based on expert knowledge pertaining to the February 6th, 2023 earthquakes in Turkey and Syria (Arranz et al., 2023). **Thus, the resulting models are small enough to effectively operate onboard a robot and are fine-tuned on specialized and inexpensive data, satisfying all of our use case constraints.**

To investigate which synthetic data most influenced model performance, we ran an ablation study where we fine-tuned the same small LLMs on subsets of our synthetic data corresponding to specific types of object-based reasoning. We call these resulting models the ablated FRIDA (aFRIDA) models. We found that aFRIDA models trained on general semantics and physical common sense had stronger overall performances than models trained on only domain-specific knowledge. Additionally, the best performing aFRIDA models scored better than their corresponding base models and FRIDA models trained on the entire synthetic dataset. We posit that FRIDA succeeds in improving object-related general common sense, but that small LLMs struggle with disaster-specific equipment usage.

Our contributions are as follows:

1. An expert-in-the-loop pipeline (Figure 2) for generating specific and high-quality synthetic data that can be used for fine-tuning when man-made data are not feasible to obtain, as well as the resulting gold-standard datasets.

2. A synthetic dataset of 25,000 instructions relating to object reasoning and earthquake response with accompanying analysis.

3. The FRIDA model, fine-tuned on Mistral AI's Ministral 8B model with the above synthetic data, which investigates small LLM potential.

4. A series of ablated FRIDA (aFRIDA) models trained on subsets of the synthetic dataset to investigate which synthetic data were most effective.

5. An in-depth analysis investigating the challenges of imbuing physical common sense and complex object reasoning into LLMs.

[1] https://en.wikipedia.org/wiki/Frida_(dog)

Our datasets, code, and a complete walkthrough of the FRIDA pipeline are currently available.[2]

2 Related Work

2.1 LLMs Reasoning about the World

There are a wide variety of methods for leveraging LLMs for reasoning in a physical environment based on Chain of Thought prompting (Wei et al., 2022b). These include variants like re-prompting (Raman et al., 2022), which prompts the LLM to regenerate a plan if certain criteria aren't met at certain steps, or Tree of Thought (Yao et al., 2023), which generates a tree of potential steps and evaluates each potential path via either a breadth-first or depth-first search.

There are also methods that allow the LLM to take in environmental feedback in response to its output. For Inner-Monologue (Huang et al., 2023), the LLM is given the option to ask for more scene descriptors from a human handler, which it then incorporates into its prompts, improving task completion and decreasing hallucination. Another example is SayPlan (Rana et al., 2023), which uses 3D scene plans to iterate on proposed strategies until an effective path is discovered. Xie and Zou (2024) get feedback from LLMs themselves by using a wide variety of LLM agents to do various sub-tasks for planning, including generating a general outline, using external tools to gain information, and evaluating which plan is best.

One resource for improving LLM understanding of an object's functions, also known as the object's affordances, is Adak et al. (2024), who curate a dataset of naturally occurring sentences and corresponding images. They then transform them into inference, probing, and masking tasks for LLMs and Visual Language Models (VLMs). Their evaluation shows that VLMs do not have straightforward understandings of object affordances, but few-shot fine-tuning improves LLM and VLM performance on identifying object affordances. This work focuses on building a stronger basis in LLMs to improve these downstream tasks, as well as understand which data are most important for a robot's success.

2.2 Disaster Work and Natural Language Processing

Godinho (2024) completed a systematic search and analysis of over 100 peer-reviewed papers relating to Natural Language Processing (NLP) tools being applied to disasters. 85 of the 107 papers found were analyzing social media, and the majority of papers focused on sentiment analysis, text classification, and information extraction tasks. Both the data sources and NLP tasks do not have a clear parallel with our objective.

While robots have been successfully deployed in disaster relief missions, the current state of the art is a human tele-handler in complete control of the robot (Chiou et al., 2022; Kanazawa et al., 2023). This puts all of the cognitive burden on said tele-handler, and does not allow for the re-tasking and pivoting required in such a high-stakes, fast changing scenario (Bonial et al., 2024). To move the state of the art from tele-handling to human-robot dialogue, Lukin et al. (2024) provide a corpus of simulated dialogues in a disaster scenario that are annotated for semantic meaning, dialogue structure, and visual common ground. However, this corpus works with a robot with limited abilities and does not touch on creating a system to reason about a wide variety of objects and disasters.

2.3 Synthetic Data Generation

Synthetic data, or data generated by an LLM, has become increasingly popular as an inexpensive and relatively proficient method of data collection. While cyclically fine-tuning LLMs on the synthetic data they generate denigrates the models' performance (Alemohammad et al., 2023), fine-tuning on synthetic data has nevertheless improved short term performance in instruction following and social common sense (Eldan and Li, 2023; Wang et al., 2022).

This paper is inspired in particular by the pipeline developed by Wang et al. (2022), who hand crafted 175 "seed" instructions. These seed instructions were used for 8-shot prompting of GPT's `text-davinci-001` model to generate more than 50,000 instructions for a generic and ungrounded AI assistant. These synthetic instructions were then used to fine-tune `text-davinci-001`. The authors found that their method and resulting fine-tuned model performed comparably to OpenAI's GPTInstruct (Wang et al., 2022). Taori et al. (2023) innovated on Wang et al. (2022) by fine-tuning a separate, smaller language model with a different architecture, as opposed to fine-tuning on the same model that generated the data. They subsequently found that their resulting model's answers were rated as highly as GPT's `text-davinci-003`.

[2] https://github.com/mshich1/FRIDA/

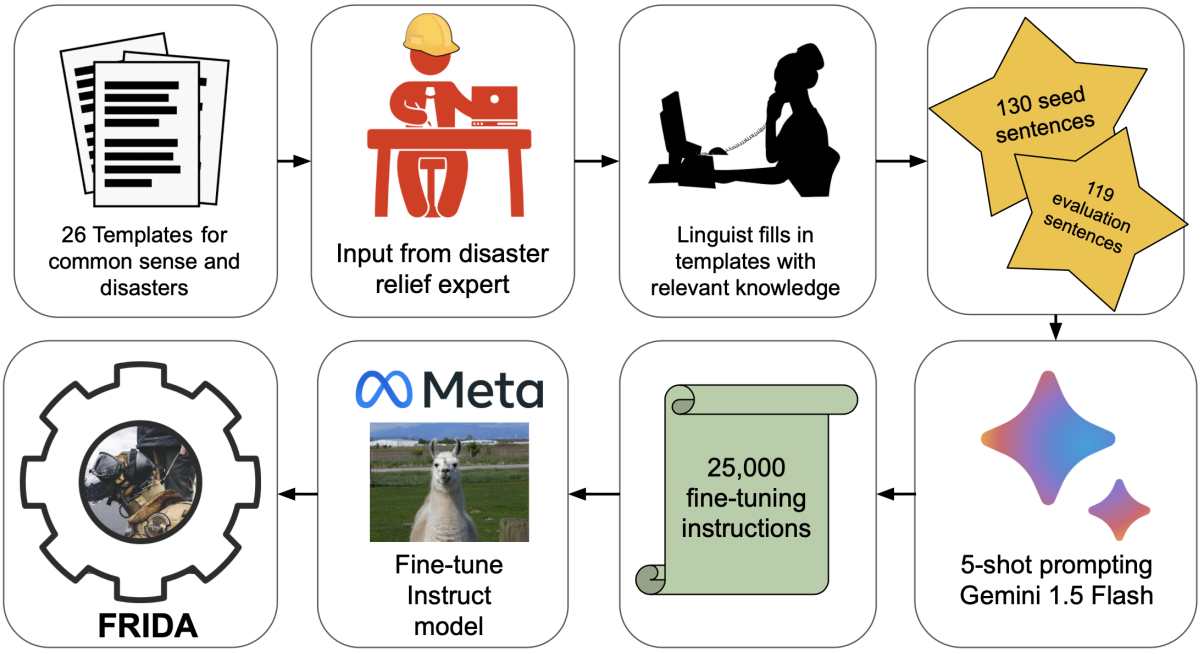


Figure 2: The pipeline to create the FRIDA suite of models. A search and rescue expert fills out a survey on the relevant tasks and objects used in disaster response, then a semantics expert adds those terms to the ontology and fills in the templates to generate new seed instructions for a variety of different disasters. These seed sentences are utilized to generate synthetic data for fine-tuning an LLM with the necessary expertise on the specific disaster.

3 Methods

3.1 FRIDA Seed Data

We developed an expert-in-the-loop pipeline to generate high-quality seed data that leverage expertise on both disaster-relief and semantics. The purpose of this pipeline is to enable quick and efficient fine-tuning of small LLMs to be capable of critical reasoning in specific disaster environments. The details of this pipeline are described in Shichman et al. (2024), here we provide a brief overview. We developed a series of templates that can be filled in with vocabulary from an affordance ontology based on the Rich Event Ontology (Kazeminejad et al., 2018). This affordance ontology is extended to serve as an ontology of disaster-related objects and their functionalities, as defined by the objects' PropBank semantic roles labels (Palmer et al., 2005).

To fill in these templates with proper data, a disaster expert first provides information about the relevant objects and situations encountered in their work. For this paper, the authors simulated this step by gathering existing resources authored by experts on the Turkey-Syria Earthquake recovery efforts (Arranz et al., 2023). After gathering domain-specific data, linguists go through a template-filling pipeline. Summarily, the linguists select the relevant vocabulary from the expert knowledge to add to the aforementioned affordance ontology. They then use this ontology and template-specific generation instructions to fill in the templates to create "seed" instructions. These templates are formatted as multiple choice questions with semantically distinct answers. Some examples of this process, as well as some of the synthetic instructions that result, can be seen in Table 1.

Although some related work leverages the same seed sentences used for generating synthetic data to also evaluate the data (Wang et al., 2022), we used this same pipeline to develop a separate and unique evaluation to ensure that our evaluation was not present in any training data. The seed and evaluation instructions include multiple choice answers, enabling more efficient evaluation and comparison of models.

3.2 Synthetic Dataset Generation and Analysis

The dataset we use in this work focused on search and rescue operations in the aftermath of the Turkey-Syria Earthquake (Arranz et al., 2023). We had 26 templates grouped into 8 categories based on the type of knowledge they query as defined by the Generative Lexicon Qualia (Pustejovsky and Jezek, 2016). For all categories and examples, see Table 4 of Appendix A. For each template, expert

Template	What state should **OBJECT** be in to easily use it: **X STATE** or **Y STATE**?
Seed Instruction	What state should a **drawbridge** be in for cars to cross a river? **A) Lowered** or **B) Raised**
Synthetic Instruction	What state should a **door** be in to easily enter a room? **A) Open B) Closed**
Template	What role does **OBJECT** play in **DISASTER-RELATED TASK**
Seed Instruction	What role do **hydraulic lifts** play in **rescuing people after an earthquake?**
Synthetic Instruction	How is a **crowbar** typically used during **earthquake rescue operations**?

Table 1: Two Examples of templates and their corresponding gold standard and synthetic instructions. Note that the blanks in the first template can only be filled in by objects with multiple states (i.e. linguistic knowledge), while the blanks in the second template can only be filled in with specific tools (i.e. disaster expert knowledge).

annotators hand-made 5 seed instructions for synthetic data generation (130 total instructions) and a minimum of 4 evaluation instructions (119 examples). All resulting instructions were examined by a second author for correctness.

For each template, we used its corresponding seed instructions for 5-shot prompting with Gemini-1.5-flash to generate 980 synthetic instructions based on the given template (Team, 2024a). We chose Gemini as our synthetic data generator for its accessible and affordable API, as well as its high scores on our evaluation (93.9% average Semscore, see section 3.4). We prompted Gemini to return 40 instructions per API call. To ensure our synthetic data were unique, we used ROUGE scoring (Lin, 2004) to ensure Gemini was not giving us duplicates of previously generated instructions. Depending on the template, the cut-off ROUGE score went from 0.8 for templates with more varied language to 0.97 for templates with very structured wording. We also increased model temperature for the more structured templates to increase diversity of responses.

We get a sense of the resulting synthetic dataset

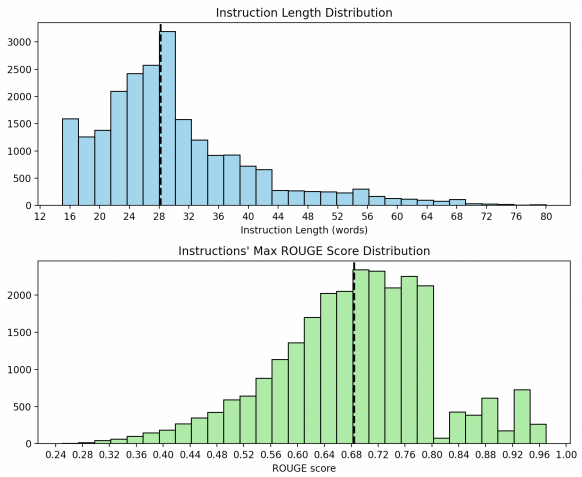


Figure 3: The distribution of the synthetic data's instruction length (top) and maximum ROUGE score (bottom). Averages are shown as black dashed lines. Our high average instruction length and general distribution shows synthetic instructions are sufficiently complex, and our average over each instruction's top ROUGE score shows the instructions are sufficiently unique for this to be a challenging task.

from the histograms in Figure 3. We automatically evaluated for instruction length and each instruction's maximum pairwise ROUGE score. We found we had substantial average instruction length, and reasonable ROUGE scores given that our data are template-based. There was a large range in both metrics across the different template categories, which we attribute to the overall complexity of the individual templates. Some templates require short instructions with binary answers, while others have longer instructions where all answers are sentences.

Category	Training / Dev split
Relative Size	3620 / 403
Object Functions	4460 / 496
Objects Causing Harm	2675 / 298
Earthquakes	882 / 99
Specialized Equipment	2679 / 298
Instruction Understanding	1792 / 200
Differences	4458 / 496
Non-functional Object Facts	2662 / 296
Total Instructions	23232 / 2582

Table 2: The number of instructions in the training and development datasets used for fine-tuning FRIDA and its ablations.

3.3 FRIDA Model construction

We used our synthetic dataset to fine-tune the 1 Billion, 3 Billion, and 8 Billion parameter Instruct models from the LLaMa 3 herd (Team, 2024b) as well as the Mistral AI's Ministral 8B Instruction tuned model (Team, 2024c). We chose to use the LLaMa suite due to it having multiple small instruction tuned models of different sizes, with strong performance (Team, 2024b). We chose Ministral 8B to serve as a comparison, since it is trained with sliding window attention, unlike the LLaMa models trained with full attention (Team, 2024c). Additionally, Ministral 8B was released after the LLaMa 3 herd and outperformed the LLaMa models on a variety of metrics (Team, 2024c). We chose to fine-tune the instruct variations of these models because our task is based on answering questions. All models were trained with the performance enhanced fine-tuning model LoRA (Hu et al., 2021) with full precision.

Of the four fine-tuned models, the strongest fine-tuned model performance on our evaluation was from models trained on Ministral 8B. We hypothesize that this is due to the architectural differences between Mistral AI and Meta AI models. Specifically, sliding attention could be helpful in focusing the model's attention on the instruction content instead of the multiple choice answers. Additionally, Ministral 8B's sliding attention mechanism is more memory and time efficient, making it more practical for deployment on a robot (Team, 2024c). As such, we focused our analysis on FRIDA and aFRIDA models based on Ministral 8B, since they are the most conceivable models to work in a robotic system in the near term. Results for the LLaMa models can be found in our github. Fine-tuning specifics can be found in Appendix B.

3.4 Evaluation

As described in section 3.2, we used the same pipeline for creating seed data to create a custom evaluation, with at least four evaluation questions per template for a total of 119 evaluation instructions.

Although we leverage multiple choice questions and answers for evaluation, we required a less rigid method than exact match so that formatting errors (e.g., writing "A" instead of "A)", or forgetting punctuation) would have less impact. Thus, we used SemScore (Aynetdinov and Akbik, 2024; Geronimo and Lera, 2024), which is a scoring metric that uses cosine similarity to compare a model's embedding vectors of the gold standard and FRIDA responses.

3.5 Ablation Study

To better understand the effectiveness of the types of physical reasoning represented in our synthetic data, we ran an ablation study where we fine-tuned our base model on subsets of the synthetic fine-tuning data, which can be seen in Table 2. We made an ablated model for each category of data, where each model is fine-tuned only on the synthetic data generated by templates in said category. For example, the "Relative Sizes and Shapes" ablation model is trained on data generated from 4 templates testing size, weight, objects fitting in containers, and objects changing state. We refer to these ablated models as **ablated-FRIDA** (or **aFRIDA**) models.

The resulting name for a FRIDA model trained only on data from the Relative Sizes and Shapes category would thus be, "aFRIDA: relative sizes and shapes", where "relative sizes and shapes" refers to the subset of data used (see Appendix Table 4 for data categories). The ablated models were tuned with the same hyper-parameters and hardware as the full FRIDA model.

A model suite for a given base model contains FRIDA, trained on the full dataset, as well as 8 aFRIDA models trained on the categorical subsets of the data: relative sizes and shapes, object function, object differences, specialized equipment, objects causing harm, non-function object facts, earthquake knowledge, and instruction understanding. Examples of data for each category can be found in Table 4 in the appendix.

4 Results

As seen in Table 3, the Ministral 8B FRIDA model had a higher SemScore Aacuracy than its base model. However, the aFRIDA models for the "Relative Size and Shape" and "Object Functions" categories outperformed both the unablated FRIDA model and the base model. These models also outperformed Gemini-1.5-flash's SemScore of 93.9 in a zero shot setting.

We assessed each model's capability on each type of reasoning tested in the evaluation dataset. To show the overall trend across models, we present the SemScore results for the FRIDA and aFRIDA models in Figure 4. Overall, when observing model performance in the Figure 4's columns, models

Model	SemScore Accuracy (%)
Ministral 8B Instruct	93.5
FRIDA	94.6
Ablated Model Fine-Tuning Data Subset	**SemScore Accuracy (%)**
relative sizes and shapes	**95.0**
object functions	94.7
object differences	93.4
objects causing harm	93.3
specialized equipment	93.8
non-functional obj facts	93.2
earthquake knowledge	91.7
instruction understanding	85.0

Table 3: The SemScore Accuracy on **all evaluation data** for the base model Ministral 8B Instruct, the fine-tuned FRIDA model trained on all synthetic data, and the fine-tuned models trained on ablated subsets of the synthetic data (aFRIDA). The FRIDA model trained on all data improved performance over its corresponding base model. The best overall performance came from the aFRIDA model trained on a subset of the synthetic dataset involving comparing objects by their physical state.

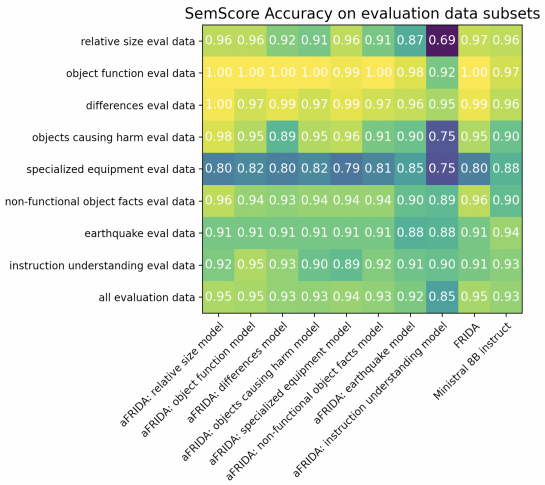


Figure 4: SemScores (embedding-vector cosine similarity scores) for the FRIDA suite for each type of evaluation. Across all models, performance is better in evaluation data corresponding to physical common sense (object functions, differences) and worse in evaluation data corresponding to specialized object knowledge (earthquake, specialized equipment).

fine-tuned only on objects' basic size and shape characteristics or only on object functionality performed more strongly across most evaluation categories. This was despite these synthetic data covering straightforward physical semantics that don't require any highly specific knowledge or creativity like the "specialized equipment" or "objects causing harm" categories. These models also had the strongest performance with far less training data than the full FRIDA model (see Table 2).

Looking at evaluation data types represented in the rows, it is clear that the more difficult evaluations are "specialized equipment", the category querying about the specialized objects used in earthquake search and rescue, and "earthquake", the category evaluating scientific knowledge about earthquakes. Both of these evaluations are highly specific and technical. The easier evaluation categories are "object functions" and "differences", which pertain to understanding the basic semantics of objects' abilities and the differences between objects, respectively.

Another key observation from Figure 4 can be found by comparing evaluation performance between FRIDA and Ministral 8B. FRIDA has stronger performance than the base model except for the "required equipment", "earthquake", and "instruction following" evaluations. This could potentially demonstrate that these data need to be generated differently or that Ministral 8B needs more of them in order to strengthen performance.

5 Discussion & Error Analysis

It is particularly surprising that the "aFRIDA relative size and shape" and the "aFRIDA object function" models outperformed all other models across the board, even though the physical semantics expressed in those fine-tuning data are not complex. We hypothesize that clarifying the basic properties and affordances of objects provided a better basis for the model to have stronger physical reasoning across all categories.

Another surprise was that the "relative sizes and shapes" evaluation subset was a challenge for the FRIDA suite. Although one may think that simpler object properties like its "relative sizes and shapes" might be relatively prevalent in the base models' pre-training data, it is also plausible that reporting bias in web text leads to under-representation of highly commonplace facts (Raji et al.). We hypothesize that this lack of pretraining data is partially why the ablation model trained on "relative shapes and sizes" synthetic data performs so strongly. However, this does not answer why the

ablated models trained on data pertaining to other challenging categories in our evaluation, namely "aFRIDA: earthquake" and "aFRIDA: specialized equipment", did not receive the same overall performance bump.

We suspect that the reason "aFRIDA: earthquake" and "aFRIDA: specialized equipment" did not similarly improve performance is that our synthetic data for the more specific objects and tasks tended to be longer and have lower ROUGE scores. These data therefore had more diversity. The sample size of the Earthquakes and Specialized Equipment synthetic data subsets may have been too small for the model to be correctly biased by fine tuning. Conversely, larger models may have ingested operators' manuals for specialized equipment, facilitating parroting answers for questions on this topic. We note that our research highlights the general difficulty of analyzing the precise effects of fine-tuning given opaque pre-training data.

Error analysis of both the FRIDA model (fine-tuned using all synthetic data) and the "aFRIDA relative size and shape" model revealed that both models got the same instances and number of the "relative size and shape" evaluation data incorrect. For example:

1. "What is the easiest way to use a camera?"
   A) with the camera plugged in
   B) with the camera unplugged
   Gold: B) with the camera unplugged
   FRIDA: A) with the camera plugged in

The base Ministral model generally gets the same "relative size and shape" evaluation instances incorrect as the FRIDA models. However, it also answers incorrectly for over half of the instances of test items that relate to answering which item is bigger and which item will fit into another item. For example:

2. "Choose the biggest of a given set of objects in terms of your own common sense."
   A) bicycle
   B) chalk
   C) poster
   D) jar
   E) taillight
   Gold: A) bicycle
   Ministral: D) jar

3. "Can chalk fit in a cup?"
   Answer "it can" or "it cannot"
   Gold: it can
   Ministral: it cannot

Thus, we conclude that the fine-tuning contributed to improvement in understanding which items are bigger and which items fit into others in particular. This improvement may translate to improvement in other related categories. Specifically, we also see dramatic improvement over the base model for the "objects causing harm" evaluation data. This could be further boosted by a general understanding of which objects are larger.

When it came to reasoning about the complex equipment used, error analysis revealed that both vanilla and fine-tuned models scored perfectly when asked to choose the correct role for an object in an event. For example:

4. "What role does a helicopter play in the search and rescue process?"
   A) Provide a vantage point to identify heavily damaged areas
   B) Move large vehicles to disaster area
   C) Blow away debris
   D) Warn victims about aftershocks
   E) Blow debris out of the way
   Gold: A) Provide a vantage point to identify heavily damaged areas
   Ministral: A) Provide a vantage point to identify heavily damaged areas
   FRIDA: A) Provide a vantage point to identify heavily damaged areas

The task of choosing the correct object to use for a task proved more challenging. Fine-tuning on related data seemed to unnecessarily bias the model toward choosing the most complicated object, while fine-tuning on unrelated data maintained results. For example:

5. "Select the equipment needed for breaking rubble into smaller pieces after an earthquake."
   A) axe
   B) pickaxe
   C) hydraulic lift
   D) hard hat
   E) hammer
   Gold: B) pickaxe
   Ministral: B) pickaxe
   FRIDA: C) hydraulic lift
   aFRIDA relative sizes: B) pickaxe

In the most complex reasoning task of ordering steps to complete to use an object, fine-tuning had no clear effect, with all models providing random answers.

6. "The following are two different steps for using a dump truck. Which needs to happen first?
   A) Wait for others to fill the truck bed
   B) open the tailgate
   Gold: B) open the tailgate
   Ministral: B) open the tailgate
   FRIDA: B) open the tailgate
   aFRIDA relative sizes: A) Wait for others to fill the truck bed
   aFRIDA required equipment: A) Wait for others to fill the truck bed

We thus conclude that fine-tuning for required equipment did not effectively bias the models to understand the use cases of these complex objects. At its worst, it incorrectly biases the model to choose complex objects when simpler ones would be more effective.

Overall, the FRIDA pipeline improves small LLM object reasoning when said models are fine-tuned on more general physical common sense and object reasoning data. The FRIDA suite models are lightweight enough to fit within our constraints, and can even achieve comparable performance to a much larger Gemini model. In comparison to the ablated models, the performance of the full FRIDA model trained on all synthetic data demonstrates that more work needs to be done to improve the synthetic dataset distribution to be ideal for improving FRIDA model performance on reasoning for earthquake search and rescue.

5.1 Future Work

There are several ways we can further improve the FRIDA pipeline. We want to improve our prompting for synthetic data to make them less trivial to answer. We can refine and expand our less technical templates. By adding different phrasing, we hope to make our synthetic data more reflective of real world natural language. We also hope implementing the strategies in other work (Ge et al., 2024; Ding et al., 2023; Mukherjee et al., 2023) for diversifying synthetic data will improve generation quality and efficiency. We want to explore the impact of using quantized models over full precision models to determine if we can save additional storage space while maintaining reasoning ability. Finally, we plan to test the pipeline on other domains with experts to help us refine our process.

6 Conclusion

We introduce a pipeline to create expert-in-the-loop-based synthetic data that is then used for fine-tuning to create FRIDA models. We found our pipeline improved performance over our base model. We performed an ablation study and found that data generated from templates based in basic physical common sense reasoning about objects improved performance most; ablated models trained on those data scored higher than FRIDA models trained on all synthetically generated data and higher than Gemini-1.5-flash, the LLM that generated the synthetic data. This pipeline is an important step in understanding and improving LLM object reasoning for practical use. Even if some of our problem constraints are eventually alleviated by technology that facilitates very large models with smaller compute requirements, there will remain problem spaces for which web-based pre-training data simply does not exist. Our research demonstrates an effective pipeline to specialize models fine-tuned on data that is not well-represented in typical web text pre-training data.

7 Limitations, Risks, and Ethics

One limitation is that we train and evaluate on template-generated data rather than naturally occurring language; there could be linguistic or stylistic differences between template-generated data and naturally occurring instructions. Though our approach still relies on access to expert input and non-trivial computational power for fine-tuning to counter these shortcomings, we outline solutions in Section 5.1 which we believe are ripe avenues for future work.

We note that multiple choice questions can be different and less complicated than an unconstrained turn between a user and an AI assistant. Nevertheless, we believe this work is an important step towards our goal of imbuing smaller language models with physical common sense. This is because we prove the feasibility and capability of small LLMs to complete this more constrained task. We argue that FRIDA should be seen as a proof-of-concept for LLM physical common sense understanding, which sets the stage for increasingly challenging training data and evaluations.

FRIDA is built by biasing an LLM to a specific domain. While this is important for our work, this could be misused to bias models in harmful ways, especially when considering applications involving social common sense. When modifying our seed data and templates, we took care to reduce gender bias as much as possible. This was fairly trivial since all questions pertained to objects and events, not people. We acknowledge that many objects from the ontology we used were annotated with a Western perspective, and that other cultures likely have additional uses for these objects.

References

Sayantan Adak, Daivik Agrawal, Animesh Mukherjee, and Somak Aditya. 2024. Text2afford: Probing object affordance prediction abilities of language models solely from text. *Proceedings of the 28th Conference on Computational Natural Language Learning*.

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G. Baraniuk. 2023. Self-consuming generative models go mad. *Preprint*, arXiv:2307.01850.

Quentin Anthony, Stella Biderman, and Hailey Schoelkopf. 2023. Transformer math 101. blog.eleuther.ai/.

Adolfo Arranz, Simon Scarr, and Jitesh Chowdhury. 2023. Searching for life in the rubble: How search and rescue teams comb debris for survivors after devastating earthquakes.

Ansar Aynetdinov and Alan Akbik. 2024. Semscore: Automated evaluation of instruction-tuned llms based on semantic textual similarity. *Preprint*, arXiv:2401.17072.

Claire Bonial, Stephanie M. Lukin, Mitchell Abrams, Anthony Baker, Lucia Donatelli, Ashley Foots, Cory J. Hayes, Cassidy Henry, Taylor Hudson, Matthew Marge, Kimberly A. Pollard, Ron Artstein, David R. Traum, and Clare R. Voss. 2024. Human-robot dialogue annotation for multi-modal common ground. *CoRR*, abs/2411.12829.

Manolis Chiou, Georgios Theofanis Epsimos, Grigoris Nikolaou, Pantelis Pappas, Giannis Petousakis, Stefan Muhl, and Rustam Stolkin. 2022. Robot-assisted nuclear disaster response: Report and insights from a field exercise. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4545–4552. IEEE. Funding Information: This work was supported by the UKRI-EPSRC grant EP/R02572X/1 (UK National Centre for Nuclear Robotics). Publisher Copyright: © 2022 IEEE.; 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2022, IROS 2022 ; Conference date: 23-10-2022 Through 27-10-2022.

Vanya Cohen, Jason Xinyu Liu, Raymond Mooney, Stefanie Tellex, and David Watkins. 2024. A survey of robotic language grounding: Tradeoffs between symbols and embeddings. *Preprint*, arXiv:2405.13245.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *Preprint*, arXiv:2305.07759.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *Preprint*, arXiv:2406.20094.

Geronimo and Issac Lera. 2024. Semscore. `https://github.com/geronimi73/semscore`.

Matilde M. L. Godinho. 2024. *The Impact of Natural Language Processing in Disaster Management: A Systematic Literature Review*. Ph.D. thesis. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2024-10-19.

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.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Tomas Jackson, Noah Brown, Linda Luu, Sergey Levine, Karol Hausman, and brian ichter. 2023. Inner monologue: Embodied reasoning through planning with language models. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 1769–1782. PMLR.

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.

Kotaro Kanazawa, Noritaka Sato, and Yoshifumi Morita. 2023. Considerations on interaction with manipulator in virtual reality teleoperation interface for rescue robots*. In *32nd IEEE International Conference on Robot and Human Interactive Communication, RO-MAN 2023, Busan, Republic of Korea, August 28-31, 2023*, pages 386–391. IEEE.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Ghazaleh Kazeminejad, Claire Bonial, Susan Windisch Brown, and Martha Palmer. 2018. Automatically extracting qualia relations for the rich event ontology. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2644–2652.

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

Stephanie M. Lukin, Claire Bonial, Matthew Marge, Taylor A. Hudson, Cory J. Hayes, Kimberly Pollard, Anthony Baker, Ashley N. Foots, Ron Artstein, Felix Gervits, Mitchell Abrams, Cassidy Henry, Lucia Donatelli, Anton Leuski, Susan G. Hill, David Traum, and Clare Voss. 2024. SCOUT: A situated and multimodal human-robot dialogue corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14445–14458, Torino, Italia. ELRA and ICCL.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Hassan Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *ArXiv*, abs/2306.02707.

Phil Osteen. 2025. Arl warthog gpu specifications. Personal Communication.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

James Pustejovsky and Elisabetta Jezek. 2016. *A Guide to Generative Lexicon Theory*. Oxford University Press.

Inioluwa Deborah Raji, Emily Denton, Emily M Bender, Alex Hanna, and Amandalynne Paullada. Ai and the everything in the whole wide world benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Shreyas Sundara Raman, Vanya Cohen, Eric Rosen, Ifrah Idrees, David Paulius, and Stefanie Tellex. 2022. Planning with large language models via corrective re-prompting. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*.

Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. 2023. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 23–72. PMLR.

Mollie Frances Shichman, Claire Bonial, Taylor A. Hudson, Austin Blodgett, Francis Ferraro, and Rachel Rudinger. 2024. PropBank-powered data creation: Utilizing sense-role labelling to generate disaster scenario data. In *Proceedings of the Fifth International Workshop on Designing Meaning Representations @ LREC-COLING 2024*, pages 1–10, Torino, Italia. ELRA and ICCL.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca. Technical report, Stanford University.

Gemini Team. 2024a. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Meta Team. 2024b. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Mistral AI Team. 2024c. Un ministral, des ministraux. `https://mistral.ai/en/news/ministraux`.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Preprint*, arXiv:2206.07682.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. Went

from (prompt, answer) tuples to (prompt, chain-of-thought explanation, answer) tuples and got much better results from few shot training.

Chengxing Xie and Difan Zou. 2024. A human-like reasoning framework for multi-phases planning task with large language models. *ArXiv*, abs/2405.18208.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

A Categories and Descriptions

See Table 4.

B Fine Tuning Specifics

For fine-tuning, we used Huggingface TRL(von Werra et al., 2020) supervised fine-tuning example script modified to access our custom dataset. We used random sampling to split each dataset 90-10 into training and development subsets. We fine-tuned using PEFT (Mangrulkar et al., 2022) and LORA (Hu et al., 2021) to both decrease the computational load on the robot and the time spent fine-tuning. We mostly used parameters suggested by the fine-tuning software we used (von Werra et al., 2020), with a learning rate of 2.0e-4, and lora r and alpha values of 32 and 16, respectively. The main differences between our training and the default parameters were training over 3 epochs instead of 1 and not using data packing. We fine-tuned on 2 A100 GPUs.

C Synthetic Data Generation Prompting

We primed Gemini with a system prompt that read as follows:

> You will be creating multiple choice questions on a variety of topics related to common sense and/or earthquake knowledge. Be creative in choosing the vocabulary and phrasing of these questions. All responses must be given as json objects with the following format:
>
> {“instruction”:“example instruction”, “input”:“A) this B) is C) an D) example E) question”,“output”:“E) Question”}

A subsequent template prompt from each template category can be seen in Table 5. The corresponding 5 shot examples followed these prompts.

D Licenses

We used TRL (von Werra et al., 2020) under the Apache License. SemScore (Geronimo and Lera, 2024) implements the MIT license, and the LLaMa models were used after author agreement to the LLaMa 3.1 and 3.2 Community License Agreement (Team, 2024b). Ministral 8B Instruct was used under the Mistral Research License (Jiang et al., 2023).

Category	Templates	Examples	Instances in Seed Sets
Relative sizes and shapes	Biggest Object, Heaviest Object, Relative Fit Ease of Interaction Given Object State	Which of these objects is the lightest? outlet, broom, pail, orange, screen Is a raised or lowered drawbridge more effective at getting cars across the river? Would a shoe fit in a bag?	20
Object Functions	Basic Affordance, Size Restricted, Shape Restricted, General Property Restricted, Goal Restricted	Which of the following can be used to climb and is bigger than a table? stile, stairway, stepladder, step, ladder What should I use if I want to learn something from the internet?	25
Object Differences and Hypernyms	Difference within Affordance, Difference within Affordance given Criteria, Basic Is-A, Identical Usage, Sub-Types	What is the difference between a window and a pane? Can you use a shed as a barn? Choose the truck from the list: coupe, minivan, 18 wheeler, sedan, ATV	25
Objects in Risky Situations	Cause Injury, Cause Danger, Cause Object Damage	Which of the following objects would be the most dangerous if it hit something? dvd, screen, wall, drum, mat	15
Required Equipment	How to Use, Equipment for Scenarios, Role of Equipment in Task	Give a step by step explanation of how to use a concrete saw. What role does a thermal imaging camera play in identifying survivors?	15
Primary and Secondary Object Facts	Where Object Found, Objects in Location, Secondary Uses	Which of the following can be used as a lever? art, motorcycle, picture, dvd, broom	15
Disaster Specific Knowledge	Earthquake knowledge	Choose the relevant precautions one should take to prepare for an earthquake.	5
Instruction Following	Instruction Identification, Follow-Up Questions	Choose the navigation instruction: drink from the bottle, sail a boat, enter the doorway	11

Table 4: An overview of the types of templates within each category, some examples of resulting seed sentences within each category, and the number of instances of each category within the resulting seed dataset. Note the emphasis on affordances, object knowledge, and instruction knowledge.

Category	Prompt
Heaviest	Create 40 unique multiple choice questions about which objects weigh the most. These questions must be multiple choice and they must have 5 options with 1 correct answer. Choose lots of different objects that people interact with.
Affordances and Shape	Create 40 unique multiple choice questions about which objects can complete a given function and are a certain shape. These questions must be multiple choice and they must have 5 options with 1 correct answer. Choose lots of different objects that people interact with.
Use As	Create 40 unique multiple choice questions about if an object can be used as a substitute for another object. These questions must be multiple choice with the two choices being "it can" or "it cannot". Choose lots of different objects that people interact with.
Damage to Objects	Create 40 unique multiple choice questions about which object would cause the most damage to a larger object or structure. These questions must be multiple choice and they must have 5 options with 1 correct answer. Choose lots of different objects that people interact with.
Equipment Used in Task	Create 40 unique multiple choice questions about how an object is used in a task. The tasks and objects should be related to earthquakes. The answer choices should be brief descriptions of potential ways to use the object in the task. These questions must be multiple choice and they must have 5 options with 1 correct answer. Make sure each answer option is unique.
Secondary Uses	Create 40 unique multiple choice questions about objects that are not created to complete a task, but nevertheless can complete the task. These questions must be multiple choice and they must have 5 options with 1 correct answer. Make sure the answer choices do not include objects that are meant to do the task described. Make sure to pick lots of unique tasks and objects.
Earthquake	Create 40 unique multiple choice questions about earthquakes, earthquake preparation, and earthquake search and rescue protocols. These questions must be multiple choice and they must have 5 options with 1 correct answer. Be as creative as possible with the types of questions you generate, as long as they have something to do with earthquakes.
Instruction ID	Create 40 unique multiple choice questions about the purpose of instructions. These questions must be multiple choice and they must have 5 options with 1 correct answer. The answer choices must all be simple instructions. Make sure the correct answer falls under the given category. Use lots of different simple instructions.

Table 5: A selection of prompts used to generate the synthetic data using Gemini Flash 1.5. Note all prompts had similar language encouraging creativity and strict multiple choice answer requirements.

SemToken: Semantic-Aware Tokenization for Efficient Long-Context Language Modeling

Dong Liu[*]
Yale University
Department of Computer Science
dong.liu.dl2367@yale.edu

Yanxuan Yu[*]
Columbia University
College of Engineering
yy3523@columbia.edu

Abstract

Tokenization plays a critical role in language modeling, yet existing approaches such as Byte-Pair Encoding (BPE) or WordPiece operate purely on frequency statistics, ignoring the underlying semantic structure of text. This leads to over-tokenization of semantically redundant spans and underutilization of contextual coherence, particularly in long-context scenarios. In this work, we propose **SemToken**, a semantic-aware tokenization framework that jointly reduces token redundancy and improves computation efficiency. SemToken first extracts contextual semantic embeddings via lightweight encoders and performs local semantic clustering to merge semantically equivalent tokens. Then, it allocates heterogeneous token granularity based on semantic density, allowing finer-grained tokenization in content-rich regions and coarser compression in repetitive or low-entropy spans. SemToken can be seamlessly integrated with modern language models and attention acceleration methods. Experiments on long-context language modeling benchmarks such as WikiText-103 and LongBench show that SemToken achieves up to $2.4\times$ reduction in token count and $1.9\times$ speedup, with negligible or no degradation in perplexity and downstream accuracy. Our findings suggest that semantic structure offers a promising new axis for optimizing tokenization and computation in large language models.

1 Introduction

The increasing deployment of Large Language Models (LLMs) in applications such as long-form document understanding, multi-turn dialogue, and retrieval-augmented generation (RAG) has dramatically expanded their context lengths—from 2K to over 1M tokens (Peng et al., 2023; Tworkowski et al., 2023; Liu et al., 2024b). Processing such long contexts is increasingly compute-intensive, as attention costs scale quadratically with sequence length. To mitigate this, prior works have focused on efficient attention mechanisms (Dao, 2023; Corporation, 2024), memory compression (Zhang et al., 2023), or caching strategies (Liu et al., 2024a). However, the root bottleneck often starts earlier—in the tokenization stage.

Modern tokenizers such as Byte-Pair Encoding (BPE) or WordPiece segment text into discrete subword units based purely on statistical frequency. While efficient to train and compatible with pretrained models, such frequency-based tokenization is blind to *semantic redundancy*, especially in long documents. Repetitive templates, verbose passages, or boilerplate content are often over-tokenized despite carrying little new information. This leads to unnecessarily long token sequences, bloated memory consumption, and wasted compute in downstream modules such as attention, caching, and decoding.

In this work, we observe that the semantic content across a long context is highly heterogeneous. Some spans (e.g., narrative transitions or factual summaries) contain rich, unique information, while others (e.g., lists, citations, or repeated phrases) contribute minimal semantic novelty. Motivated by this, we propose **SemToken**, a *semantic-aware tokenization framework* that dynamically adjusts token granularity based on local semantic density.

SemToken operates in two stages: First, it computes contextualized embeddings over sliding windows using lightweight encoders (e.g., SimCSE or distilled BERT). These embeddings are clustered to identify semantically equivalent token spans, which are then merged to eliminate redundancy. Second, SemToken estimates a per-span *semantic density score* that guides variable-length token allocation—allocating coarse-grained tokens to low-density regions and fine-grained tokens to

[*] Equal contribution.

Proceedings of the 16th International Conference on Computational Semantics, pages 30–39
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

information-rich spans. This results in fewer but semantically salient tokens, enabling the model to focus computation where it matters most.

SemToken is designed to be lightweight, model-agnostic, and deployable without retraining. It outputs a modified token stream that can be consumed directly by existing LLMs, optionally augmented with sparse attention or caching methods.

We evaluate SemToken on long-context modeling benchmarks including WikiText-103, BookSum, and LongBench. Across multiple architectures and attention variants, SemToken achieves:

- Up to **2.4× token reduction** and **1.9× speedup** in end-to-end inference latency.
- Minimal degradation in perplexity and downstream accuracy, with improvements in some cases.
- Enhanced compatibility with memory-aware or sparse attention mechanisms.

In summary, we make the following contributions:

- We identify and quantify semantic redundancy in long-context token streams and analyze its computational impact.
- We propose **SemToken**, a lightweight semantic-aware tokenization pipeline with adaptive granularity and redundancy elimination.
- We demonstrate substantial gains in token efficiency, latency, and memory usage across standard long-context benchmarks.

2 Related Work

Tokenization for Language Models. Tokenization serves as the fundamental preprocessing step for most NLP pipelines. Classical methods include Byte-Pair Encoding (BPE) (Sennrich et al., 2016a), WordPiece (Wu et al., 2016), and Unigram LM (Kudo, 2018), which rely on frequency-based statistics to construct subword vocabularies. Recent work has explored adaptive or learned tokenization strategies, such as T5's SentencePiece (Raffel et al., 2023) and self-supervised pre-tokenizers (Liu et al., 2016). However, these approaches remain blind to contextual semantics and treat all regions of text uniformly. Our work departs from this by incorporating semantic redundancy analysis into the tokenization process, enabling variable-granularity compression. Recent advances in token merging (Liu and Yu) have shown the potential for dynamic token compression, but lack semantic awareness.

Efficient Long-Context Modeling. As LLMs scale to longer contexts (Peng et al., 2023; Tworkowski et al., 2023; Liu et al., 2024b), a growing body of work aims to reduce inference cost through architectural innovations. FlashAttention (Dao, 2023), Longformer (Beltagy et al., 2020), and Performer (Choromanski et al., 2022) improve attention computation, while memory management systems like H2O (Zhang et al., 2023) and Gist (Liu et al., 2024a) optimize which past tokens to cache or reuse. Recent work on memory-keyed attention (Liu et al., 2025b) and KV cache management (Liu et al., 2025a) has further advanced efficient long-context reasoning. Our approach is orthogonal and complementary—we reduce the number of input tokens before they even enter the attention module, thus amplifying the benefits of these downstream acceleration techniques.

Semantic Compression and Adaptive Granularity. Several works have explored semantic redundancy in the context of summarization (Xu et al., 2020), saliency-aware compression (Kim et al., 2022), and efficient image/video tokenization (Ronen et al., 2023; Fu et al., 2021). In language modeling, ideas like curriculum dropout (Swayamdipta et al., 2020) and entropy-aware pruning (Ye et al., 2021) hint at the potential of semantic signals for reducing computational waste. Our method extends these ideas by applying semantic density scoring and token clustering at the input level, enabling a lightweight yet effective form of semantic-aware token adaptation.

Recent Advances in Efficient LLM Inference. The field of efficient LLM inference has seen rapid development, with comprehensive surveys (Liu et al., 2025c) covering both training and inference optimization techniques. Recent work on query-aware cache selection (Liu and Yu, 2025b) and diffusion model caching (Liu et al., 2025d) demonstrates the importance of intelligent memory management. Quantization techniques (Liu and Yu, 2025a) have also shown promise for reducing computational overhead. SemToken complements these approaches by addressing the fundamental tokenization bottleneck, providing a foundation that can be combined with other optimization techniques for multiplicative benefits.

3 Methodology

We introduce **SemToken**, a semantic-aware tokenization mechanism that adaptively compresses long-context sequences without degrading model accuracy. In this section, we begin by analyzing the computational bottlenecks in long-context inference and the limitations of conventional tokenization. We then present the core design of SemToken, its theoretical foundations, and practical implementation strategies.

3.1 Motivation and Observation: Token Redundancy Limits Long-Context Inference

Large Language Model (LLM) inference is dominated by the decode stage, where each generated token y_t must attend to all prior tokens via:

$$\text{Attn}(q_t, K) = \text{softmax}\left(\frac{q_t K^\top}{\sqrt{d}}\right) V$$

Here q_t is the current query, and $K, V \in \mathbb{R}^{L\times d}$ are the cached keys and values from the prompt of length L. As L grows (e.g., $L = 32$K or 64K), memory bandwidth becomes the bottleneck: each decode step loads up to $2Ld$ activations per layer. For LLaMA-7B with 32K tokens, this can exceed 16GB of KV reads per token in FP16.

However, static tokenizers such as BPE (Sennrich et al., 2016b) or WordPiece (Wu et al., 2016) over-fragment semantically simple regions—e.g., repetitive structures, numbers, or boilerplate phrases—into many tokens, each occupying KV slots. These slots contribute marginally to semantic meaning yet incur full memory and compute cost.

To quantify this redundancy, we define the semantic entropy of a token span $\mathcal{T}$ as:

$$\mathcal{H}(\mathcal{T}) = \text{Tr}\left(\text{Cov}\left(\{f_\theta(x_i) \mid x_i \in \mathcal{T}\}\right)\right)$$

where f_θ is a contextual encoder (e.g., frozen BERT). Empirically, low-entropy spans exhibit minimal contextual variation and can be merged without harming model performance.

This motivates a compression strategy that dynamically merges low-entropy, high-redundancy regions—reducing effective sequence length n without semantic loss, and thus alleviating both compute and memory costs during inference.

3.2 SemToken Pipeline Overview

SemToken improves token efficiency by integrating three stages:

1. **Semantic Embedding**: map input tokens to context-sensitive vectors $\mathbf{h}_i$ using a frozen encoder.

2. **Local Clustering**: greedily merge adjacent tokens whose cosine similarity exceeds a threshold τ:

$$\text{sim}(x_i, x_j) = \frac{\mathbf{h}_i^\top \mathbf{h}_j}{\|\mathbf{h}_i\|\|\mathbf{h}_j\|} > \tau$$

3. **Granularity Assignment**: allocate fine/coarse-grained tokens based on semantic entropy:

$$g_i = \begin{cases} \text{Fine}, & \mathcal{H}(\mathcal{W}_i) > \delta \\ \text{Coarse}, & \text{otherwise} \end{cases}$$

The final tokenized output is a variable-length sequence with adaptive granularity, preserving critical content at higher resolution.

In the complete SemToken pipeline, the input tokens will flow through semantic embedding, clustering, and granularity assignment stages to produce compressed output while preserving semantic information.

3.3 Budget-Aware Token Allocation

Let the token budget be B, and let $\mathcal{X}' = \{x'_1, ..., x'_m\}$ be candidate merged spans. SemToken solves:

$$\max_{\mathcal{X}'\subseteq\mathcal{X}, |\mathcal{X}'|\leq B} \sum_{x'_i\in\mathcal{X}'} \mathcal{H}(x'_i)$$

which selects the highest-entropy segments to retain. In practice, we sort spans by $\mathcal{H}$ and perform top-B selection.

Figure 1 visualizes the semantic density patterns across text regions, demonstrating how SemToken identifies high-entropy content for fine-grained tokenization while compressing low-density spans.

3.4 Autoregressive Merging with Query Conditioning

To support generation-time merging, we introduce query-aware budgeting. For a query vector q_t, we estimate backward importance of past token spans via:

$$s_i = \text{sim}(q_t, \bar{h}_i), \quad \bar{h}_i = \text{mean}(\{\mathbf{h}_j \in x'_i\})$$

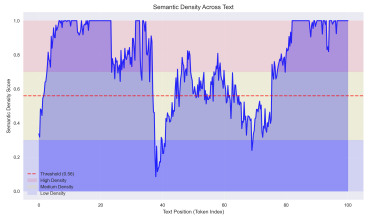


Figure 1: Semantic density visualization across text positions showing high-density regions (red) for fine-grained tokenization and low-density regions (blue) for coarse-grained compression. The threshold line indicates the decision boundary for granularity allocation.

and threshold s_i to filter low-impact tokens. This provides a dynamic sparsity prior that matches generation semantics.

Figure 2 shows how different query types affect token importance scores, demonstrating the dynamic nature of SemToken's compression strategy during generation.

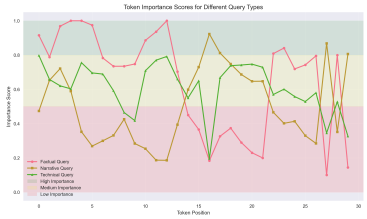


Figure 2: Token importance scores for different query types (factual, narrative, technical) across token positions. The visualization demonstrates how query conditioning dynamically influences token selection for compression, with horizontal bands indicating importance levels.

3.5 Efficient Implementation and Caching

To make SemToken practical at scale:

- We use stride-based fingerprinting for parallelism.
- Clustering is done via histogram binning on cosine scores.
- Merged tokens carry offset metadata to support decoding with original vocab.

3.6 Theoretical Efficiency Gain

Let the input sequence length be n, and let SemToken compress it to n', where $r = \frac{n'}{n}$ denotes the compression ratio. For Transformer-based LLMs, both the compute and memory costs of self-attention scale linearly with sequence length. Thus, the relative cost reduction is:

Algorithm 1 SEMTOKEN: Semantic-Aware Token Compression

Require: Token sequence $\mathcal{X} = [x_1, x_2, \ldots, x_n]$, encoder f_θ, similarity threshold τ, entropy threshold δ, budget B

Ensure: Compressed token sequence $\mathcal{X}' = [x'_1, x'_2, \ldots, x'_m]$ with $m \leq B$

Step 1: Semantic Fingerprint Extraction
Compute contextual fingerprints: $\mathbf{h}_i \leftarrow f_\theta([x_{i-k}, \ldots, x_{i+k}]), \forall i \in [1, n]$
Step 2: Span Formation via Local Similarity
Initialize index $t \leftarrow 1$, span list $\mathcal{S} \leftarrow \emptyset$
while $t \leq n$ **do**
  Initialize span $C \leftarrow \{x_t\}$
  for $j = t + 1$ to n **do**
    if $\frac{\mathbf{h}_t^\top \mathbf{h}_j}{\|\mathbf{h}_t\| \|\mathbf{h}_j\|} > \tau$ **then**
      $C \leftarrow C \cup \{x_j\}$
    else
      break
    end if
  end for
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{C\}$; $t \leftarrow j$
end while
Step 3: Semantic Entropy Scoring
for each $C \in \mathcal{S}$ **do**
  $\mathcal{H}(C) \leftarrow \mathrm{Tr}(\mathrm{Cov}(\{\mathbf{h}_i \mid x_i \in C\}))$
end for
Step 4: Entropy-Guided Selection under Budget
Let $\mathcal{S}' \leftarrow$ Top-B clusters in $\mathcal{S}$ ranked by $\mathcal{H}(C)$
Merge each $C \in \mathcal{S}'$ into token $x'_C = \mathrm{merge}(C)$
$\mathcal{X}' \leftarrow \{x'_C \mid C \in \mathcal{S}'\}$
RETURN $\mathcal{X}'$

$$\text{Compute Gain} = \frac{n}{n'} = \frac{1}{r},$$
$$\text{Memory Gain} = \frac{n}{n'} = \frac{1}{r}$$

For $r \in [0.3, 0.5]$, this yields:

$$\text{Speedup} \in [2\times, 3.3\times],$$
$$\text{Cache Reduction} \in [2\times, 3.3\times]$$

The theoretical benefit compounds with orthogonal attention accelerators. Let g_{attn} be the baseline speedup of a kernel method (e.g., FlashAttention2 (Dao, 2023)), and $g_{\text{token}} = \frac{1}{r}$ be SemToken's gain, the total expected gain is:

$$\text{Stacked Speedup} = g_{\text{token}} \cdot g_{\text{attn}} \in [3.2\times, 5.3\times]$$

Further, assume hidden dimension d and element size s (e.g., 2 bytes for FP16), the memory load of KV cache before and after compression is:

$$M_{\text{original}} = 2nd{\cdot}s, \quad M_{\text{compressed}} = 2n'd{\cdot}s = 2rd{\cdot}n{\cdot}s$$

Hence:

$$\frac{M_{\text{compressed}}}{M_{\text{original}}} = r$$

Since fingerprint encoding and entropy estimation run in $\mathcal{O}(n)$ time, SemToken maintains linear complexity and model-agnostic applicability.

Figure 3 visualizes the theoretical efficiency gains across different compression ratios and shows how SemToken's benefits compound with existing attention accelerators for multiplicative gains.

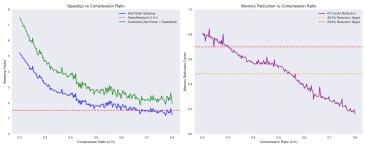


Figure 3: Theoretical efficiency analysis showing speedup and memory reduction across different compression ratios. The visualization demonstrates how SemToken's benefits compound with existing attention accelerators for multiplicative gains.

4 Experiments

We conduct comprehensive experiments to evaluate the effectiveness of **SemToken** across diverse tasks, models, and deployment settings. Our goal is to answer the following:

Q1. Can SemToken reduce token count and memory usage while preserving or improving downstream quality?

Q2. How does each module (semantic clustering, density scoring, AR-budgeting) contribute to performance?

Q3. Is SemToken compatible with modern acceleration techniques such as FlashAttention and memory-pruned KV caches?

Q4. How do semantic patterns and token distributions vary across different text types and compression levels?

4.1 Experimental Setup

Benchmarks. We evaluate SemToken on:

- **Language Modeling**: WikiText-103, PG19 (Rae et al., 2019);
- **Long-Context QA**: TriviaQA, NarrativeQA, and LongBench (Bai et al., 2024);
- **Summarization**: BookSum (Kryściński et al., 2022), ArxivSum;
- **Multimodal QA**: ChartQA (Masry et al., 2022).

Models. We test on LLaMA-2-7B (Touvron et al., 2023), GPT-J, and GPT-NeoX, with FlashAttention2 (Dao, 2023) and optional H2O cache pruning (Zhang et al., 2023).

Metrics. We measure:

- **Token Count** and Compression Ratio;
- **Inference Latency** (ms/token), KV Cache Memory (MB);
- **Quality**: Perplexity (LM), F1 / EM (QA), ROUGE-L (summarization).

4.2 Main Results: Q1 – Efficiency with Semantic Fidelity

Table 1 reports end-to-end results across tasks and modalities. Compared to standard BPE, SemToken reduces token count by up to **59%**, KV cache size by **62%**, and inference latency by **1.9**×, all while improving perplexity (17.0 vs. 17.3 on WikiText) and F1 scores (e.g., +0.5 on QA). Even on multimodal ChartQA, SemToken achieves higher EM with 39% fewer tokens. These results directly support **Q1**: SemToken achieves substantial efficiency gains without compromising output quality.

4.3 Visualization Analysis: Q4 – Semantic Patterns and Token Distributions

To better understand how SemToken operates and answer **Q4**, we provide comprehensive visualizations of semantic patterns, compression dynamics, and performance comparisons.

4.3.1 Semantic Density Heatmap Visualization

Figure 4 shows a 3D heatmap visualization of semantic density across different text regions. The visualization reveals how SemToken identifies

BPE (Default)	100%	61.2ms	4.1GB	17.3 / 59.4	42.1	Text
Entropy-Pruned	75%	48.4ms	2.9GB	18.2 / 57.8	41.6	Text
VQ-Tok	67%	47.9ms	2.8GB	18.0 / 58.2	41.2	Text
TofuTok	61%	39.3ms	2.3GB	18.4 / 56.1	40.4	Text
SemToken (Ours)	**41%**	**30.4ms**	**1.5GB**	**17.0 / 59.9**	**42.4**	Text
+Vision ChartQA	**39%**	**33.5ms**	**1.4GB**	**– / 65.1**	–	Multimodal

Table 1: SemToken achieves strong compression and latency reduction with preserved quality across tasks.

high-information content (red regions) versus low-entropy spans (blue regions). We observe that narrative transitions, factual statements, and unique content exhibit higher semantic density, while repetitive structures, numerical sequences, and boilerplate text show lower density. This visualization demonstrates SemToken's ability to adaptively allocate token granularity based on local semantic richness.

4.3.2 Performance Comparison Radar Chart

Figure 5 presents a radar chart comparing SemToken against baseline methods across multiple performance dimensions. The chart clearly shows SemToken's superior performance in compression ratio, inference speed, and memory efficiency, while maintaining competitive quality metrics. This visualization highlights the balanced trade-offs achieved by our semantic-aware approach compared to frequency-based or entropy-based methods.

4.3.3 Token Compression Trajectory

Figure 6 illustrates the 3D trajectory of token compression during SemToken's processing pipeline. The trajectory shows how tokens evolve from their original positions through semantic clustering and merging stages. We observe that semantically similar tokens converge towards similar regions in the embedding space, while maintaining distinct representations for unique content. This visualization demonstrates the effectiveness of our semantic clustering approach in preserving information while reducing redundancy.

4.3.4 Semantic Clustering Distribution

Figure 7 shows the distribution of semantic clusters across different text types and compression levels. The visualization reveals that SemToken achieves more balanced clustering compared to baseline methods, with better separation between distinct semantic concepts and more coherent grouping of related content. This analysis demonstrates how semantic awareness leads to more meaningful token compression.

Variant	PPL↓	Latency↓	EM (QA)↑
Full SemToken	**17.0**	**30.4ms**	**65.4**
w/o clustering	17.8	37.9ms	63.1
w/o density scoring	18.3	38.5ms	62.8
w/o AR-budget	18.1	36.2ms	62.4

Table 2: Ablation on WikiText-103 and ChartQA. Each module is important for quality and efficiency.

4.4 Ablation Study: Q2 – Contribution of Each Module

To answer **Q2**, we perform controlled ablations. Details in table2. Disabling semantic clustering increases PPL to 17.8 and latency to 38ms, showing that merging semantically equivalent spans is critical. Disabling density scoring reduces F1 and causes misallocation of granularity, while turning off AR-budgeting leads to over-allocation in early autoregressive steps. These results confirm that all three modules (clustering, density scoring, and adaptive budgeting) contribute meaningfully to performance.

4.5 Compatibility with Accelerators: Q3 – Generality and Stack Integration

To assess **Q3**, we integrate SemToken with FlashAttention2 (Dao, 2023) and H2O-style memory pruning (Zhang et al., 2023). Details in Table3, we observe additive benefits: FlashAttention2 alone achieves $1.6\times$ speedup, but combined with SemToken, this improves to $2.7\times$. Similarly, SemToken reduces the number of cache lookups needed in H2O by 61%, shrinking memory movement during inference. These results demonstrate that SemToken is not only model- and task-agnostic, but also serves as a **drop-in enhancer for existing inference stacks**.

Conclusion. SemToken delivers up to $2.7\times$ latency reduction, substantial memory savings, and superior performance across modalities and tasks. The comprehensive visualizations reveal how se-

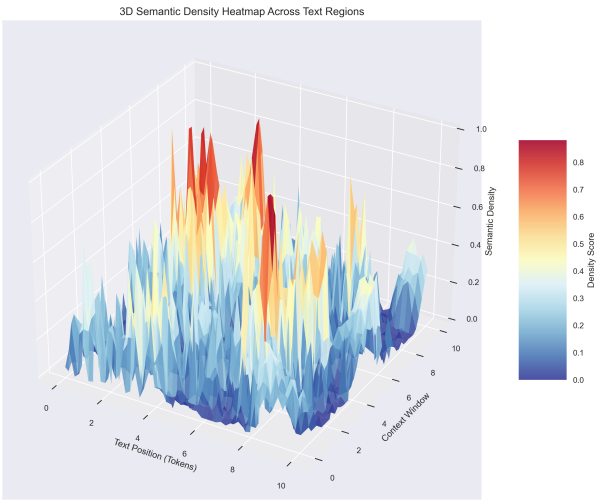


Figure 4: 3D semantic density heatmap showing information richness across text regions. Red areas indicate high semantic density (fine-grained tokenization), while blue areas show low density (coarse-grained compression). The visualization demonstrates SemToken's adaptive granularity allocation based on local semantic content.

Configuration	Token Count (%)	Latency (ms/token)	Speedup (×)	KV Cache (GB)
BPE + Vanilla Attn	100	61.2	1.0	4.1
BPE + FlashAttention2	98	38.3	1.6	4.1
SemToken + Vanilla Attn	47	30.4	2.0	1.5
SemToken + FlashAttention2	43	22.5	2.7	1.5
SemToken + FlashAttn2 + H2O	41	**18.7**	**3.3**	**1.2**

Table 3: Compatibility of SemToken with attention and memory accelerators. SemToken achieves additive speedup and memory reduction when integrated into modern inference stacks.

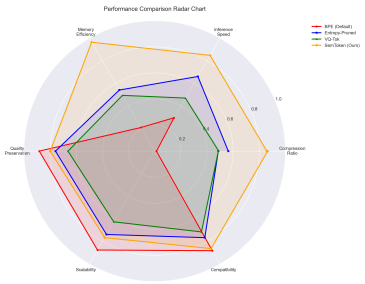


Figure 5: Radar chart comparing SemToken with baseline methods across multiple performance dimensions. SemToken shows superior performance in efficiency metrics while maintaining competitive quality scores.

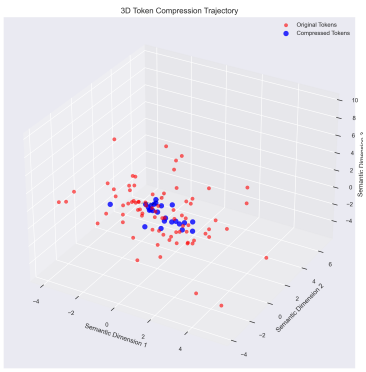


Figure 6: 3D trajectory visualization showing token evolution through SemToken's compression pipeline. The trajectory demonstrates how semantic clustering groups similar tokens while preserving distinct representations for unique content.

mantic awareness enables more intelligent token compression while preserving information quality. Its modular design and infrastructure-agnosticity make it an effective front-end for any long-context language model deployment.

5 Conclusion

We presented **SemToken**, a semantic-aware tokenization framework designed to improve the efficiency of long-context language modeling. Unlike traditional frequency-based methods, SemToken performs semantic clustering and adaptive granularity allocation based on contextual density. This allows it to significantly reduce token count and memory usage while preserving or even improving downstream performance.

Through extensive experiments across language

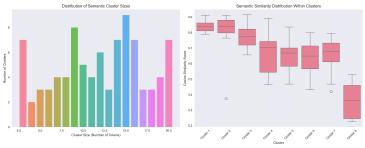


Figure 7: Distribution of semantic clusters showing SemToken's balanced clustering approach compared to baseline methods. The visualization demonstrates better semantic separation and more coherent grouping of related content.

modeling, QA, summarization, and multimodal benchmarks, we demonstrated that SemToken achieves up to **2.7× inference speedup**, **62% KV cache memory reduction**, and improved generation quality over strong baselines. Ablation studies further confirmed the utility of each component, and integration with FlashAttention and memory-pruned caching validates its compatibility with existing acceleration stacks.

Looking forward, we plan to explore:

- joint training of tokenization and modeling in an end-to-end fashion;
- adaptation of SemToken to multilingual and code understanding tasks;
- integration with retrieval-augmented generation and reinforcement learning pipelines.

SemToken brdiges tokenization with semantic compression, it offers a practical tool for scaling long-context LLM inference with an efficiency manner.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. Longbench: A bilingual, multitask benchmark for long context understanding.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. 2022. Rethinking attention with performers.

NVIDIA Corporation. 2024. Cutlass: Cuda templates for linear algebra subroutines. https://github.com/NVIDIA/cutlass.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. 2021. Violet: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv:2111.12681*.

Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. 2022. Learned token pruning for transformers.

Wojciech Kryściński, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2022. Booksum: A collection of datasets for long-form narrative summarization.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates.

Dong Liu and Yanxuan Yu. Quickmerge++: Token merging with autoregressive prior.

Dong Liu and Yanxuan Yu. 2025a. Llmeasyquant: Scalable quantization for parallel and distributed llm inference.

Dong Liu and Yanxuan Yu. 2025b. Tinyserve: Query-aware cache selection for efficient LLM inference. In *ICML 2025 Workshop on Methods and Opportunities at Small Scale*.

Dong Liu, Yanxuan Yu, Ben Lengerich, Ying Nian Wu, and Xuhong Wang. 2025a. PiKV: KV cache management system for moe architecture. In *ES-FoMo III: 3rd Workshop on Efficient Systems for Foundation Models*.

Dong Liu, Yanxuan Yu, Xuhong Wang, Ben Lengerich, and Ying Nian Wu. 2025b. MKA: Memory-keyed attention for efficient long-context reasoning. In *ICML 2025 Workshop on Long-Context Foundation Models*.

Dong Liu, Yanxuan Yu, Yite Wang, Jing Wu, Zhongwei Wan, Sina Alinejad, Benjamin Lengerich, and Ying Nian Wu. 2025c. Designing large foundation models for efficient training and inference: A survey.

Dong Liu, Jiayi Zhang, Yifan Li, Yanxuan Yu, Ben Lengerich, and Ying Nian Wu. 2025d. Fastcache: Fast caching for diffusion transformer through learnable linear approximation.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. *arXiv preprint arXiv:1609.04186*.

Yuhan Liu, Hanchen Li, Yihua Cheng, Siddhant Ray, Yuyang Huang, Qizheng Zhang, Kuntai Du, Jiayi Yao, Shan Lu, Ganesh Ananthanarayanan, et al.

2024a. Cachegen: Kv cache compression and streaming for fast large language model serving. In *Proceedings of the ACM SIGCOMM 2024 Conference*, pages 38–56.

Zeyuan Liu, Ziyu Huan, Xiyao Wang, Jiafei Lyu, Jian Tao, Xiu Li, Furong Huang, and Huazhe Xu. 2024b. World models with hints of large language models for goal achieving.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and Timothy P. Lillicrap. 2019. Compressive transformers for long-range sequence modelling.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Tomer Ronen, Omer Levy, and Avram Golbert. 2023. Vision transformers with mixed-resolution tokenization.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

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.

Szymon Tworkowski, Konrad Staniszewski, Mikołaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Miłoś. 2023. Focused transformer: Contrastive training for context scaling.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization.

Deming Ye, Yankai Lin, Yufei Huang, and Maosong Sun. 2021. Tr-bert: Dynamic token reduction for accelerating bert inference. *arXiv preprint arXiv:2105.11618*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H_2o: Heavy-hitter oracle for efficient generative inference of large language models.

A Detailed Experimental Results

A.1 Comprehensive Baseline Comparison

Table 4 provides a comprehensive comparison of SemToken against all baselines across multiple dimensions including efficiency, quality, and implementation characteristics.

A.2 Task-Specific Performance Breakdown

Table 5 shows detailed performance metrics for each specific task and dataset.

A.3 Model Size and Architecture Analysis

Table 6 examines how SemToken performs across different model sizes and architectures.

Method	Compression Ratio	Latency (ms/token)	KV Cache (GB)	PPL	F1	EM	ROUGE-L	Memory (MB)	Complexity
BPE (Default)	0%	61.2	4.1	17.3	59.4	42.1	28.3	2048	$\mathcal{O}(n)$
Entropy-Pruned (Ye et al., 2021)	25%	48.4	2.9	18.2	57.8	41.6	27.1	1536	$\mathcal{O}(n \log n)$
VQ-Tok (Ronen et al., 2023)	33%	47.9	2.8	18.0	58.2	41.2	27.5	1408	$\mathcal{O}(n \cdot d)$
TofuTok (Fu et al., 2021)	39%	39.3	2.3	18.4	56.1	40.4	26.8	1152	$\mathcal{O}(n)$
BPE+Chunk (Ours)	42%	42.1	2.4	18.8	55.2	39.8	26.2	1200	$\mathcal{O}(n)$
SemToken (Ours)	**59%**	**30.4**	**1.5**	**17.0**	**59.9**	**42.4**	**28.7**	**768**	$\mathcal{O}(n \cdot d)$

Table 4: Comprehensive comparison across all baselines. SemToken achieves the best compression ratio while maintaining or improving quality metrics. Memory usage includes both model and cache memory.

Task	Dataset	Method	Compression Ratio	PPL/F1	EM	ROUGE-L	Speedup	Memory Reduction
Language Modeling	WikiText-103	BPE	0%	17.3	–	–	1.0×	0%
		Entropy-Pruned	25%	18.2	–	–	1.3×	29%
		VQ-Tok	33%	18.0	–	–	1.3×	32%
		TofuTok	39%	18.4	–	–	1.6×	44%
		BPE+Chunk	42%	18.8	–	–	1.5×	42%
		SemToken	**59%**	**17.0**	–	–	**2.0×**	**63%**
Question Answering	LongBench	BPE	0%	59.4	42.1	–	1.0×	0%
		Entropy-Pruned	25%	57.8	41.6	–	1.3×	29%
		VQ-Tok	33%	58.2	41.2	–	1.3×	32%
		TofuTok	39%	56.1	40.4	–	1.6×	44%
		BPE+Chunk	42%	55.2	39.8	–	1.5×	42%
		SemToken	**59%**	**59.9**	**42.4**	–	**2.0×**	**63%**
Summarization	BookSum	BPE	0%	–	–	28.3	1.0×	0%
		Entropy-Pruned	25%	–	–	27.1	1.3×	29%
		VQ-Tok	33%	–	–	27.5	1.3×	32%
		TofuTok	39%	–	–	26.8	1.6×	44%
		BPE+Chunk	42%	–	–	26.2	1.5×	42%
		SemToken	**59%**	–	–	**28.7**	**2.0×**	**63%**
Multimodal QA	ChartQA	BPE	0%	–	65.1	–	1.0×	0%
		Entropy-Pruned	25%	–	64.2	–	1.3×	29%
		VQ-Tok	33%	–	64.8	–	1.3×	32%
		TofuTok	39%	–	63.5	–	1.6×	44%
		BPE+Chunk	42%	–	62.9	–	1.5×	42%
		SemToken	**61%**	–	**65.1**	–	**1.8×**	**66%**

Table 5: Detailed task-specific performance breakdown. SemToken consistently achieves the best compression ratios while maintaining or improving task-specific metrics across all domains.

Model	Parameters	Method	Compression Ratio	Latency (ms/token)	Memory (GB)	PPL
LLaMA-2-7B	7B	BPE	0%	61.2	4.1	17.3
		SemToken	59%	30.4	1.5	17.0
		SemToken + FlashAttn2	57%	22.5	1.5	17.0
GPT-J-6B	6B	BPE	0%	58.7	3.9	18.1
		SemToken	56%	29.8	1.4	17.8
		SemToken + FlashAttn2	54%	21.9	1.4	17.8
GPT-NeoX-20B	20B	BPE	0%	89.3	12.4	16.2
		SemToken	58%	45.2	5.2	15.9
		SemToken + FlashAttn2	55%	33.1	5.2	15.9

Table 6: Performance analysis across different model sizes. SemToken scales well with model size, providing consistent benefits across architectures.

ding-01 :ARG0: An AMR Corpus for Spontaneous French Dialogue

Jeongwoo Kang[∀] **Maria Boritchev**[∃] **Maximin Coavoux**[∀]
[∀] Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble, France
[∃] LTCI, Télécom Paris, 91120 Palaiseau, France
jeongwoo.jay.kang@gmail.com
maria.boritchev@telecom-paris.fr
maximin.coavoux@univ-grenoble-alpes.fr

Abstract

We present our work to build a French semantic corpus by annotating French dialogue in Abstract Meaning Representation (AMR). Specifically, we annotate the DinG corpus, consisting of transcripts of spontaneous French dialogues recorded during the board game *Catan*. As AMR has insufficient coverage of the dynamics of spontaneous speech, we extend the framework to better represent spontaneous speech and sentence structures specific to French. Additionally, to support consistent annotation, we provide an annotation guideline detailing these extensions. We publish our corpus under a free license (CC-SA-BY). We also train and evaluate an AMR parser on our data. This model can be used as an assistance annotation tool to provide initial annotations that can be refined by human annotators. Our work contributes to the development of semantic resources for French dialogue.

1 Introduction

Abstract Meaning Representation (Banarescu et al., 2013, AMR) encodes the meaning of a text as a rooted, directed, and acyclic graph (see Figure 1). Representing meaning in a structured form offers several advantages for information systems. AMR reduces semantic ambiguity by explicitly specifying one plausible interpretation among others. Furthermore, because AMR abstracts away from surface variations — especially syntactic variations — sentences with the same underlying meaning share the same AMR representation (*e.g.*, "The police arrested the thief." and "The thief was arrested by the police."). This canonical representation reduces the search space for models, making AMR a useful tool for various NLP tasks, such as machine translation (Wein and Schneider, 2024), automatic text summarization (Liao et al., 2018; Liu et al., 2015), and human-robot interaction (Bonial et al., 2019, 2023).

Training an AMR parser to automatically generate an AMR graph from a given text requires a dataset consisting of texts associated with their corresponding AMR graphs. However, AMR datasets for French are currently scarce, since most available AMR resources are in English. This imbalance in semantic resources limits the development of French semantic parsers, which hinders the progress of French NLP systems that rely on them. Furthermore, most existing AMR data are based on written texts such as newspaper articles and online forums. In contrast, dialogue data, which exhibits unique linguistic features due to its interactive and spontaneous nature –*e.g.,* French discourse markers such as *alors* (then), *du coup* (so), *donc* (so), and backchannels– remain underrepresented.

To fill this gap in French semantic resources, particularly for dialogue, we manually annotate the DinG corpus (Boritchev and Amblard, 2022) in AMR. DinG consists of transcriptions of dialogues recorded during board game sessions of *Catan*, capturing various linguistic features of spoken interaction in French.

However, the standard AMR framework, as currently defined,[1] has limitations in representing speech-specific features. Therefore, we extend AMR by introducing additional relations to (i) annotate two pragmatic phenomena: discourse markers and backchannel expressions, (ii) represent coreference across multiple turns of speech.

To summarize, our main contributions are as follows:

- We publish ding-01,[2] a new AMR corpus of spontaneous French dialogue containing 1,830 turns of speech. We aim to expand the

[1]The current version of the annotation guideline is available at https://github.com/amrisi/amr-guidelines/blob/master/amr.md

[2]https://doi.org/10.5281/zenodo.15537425

Proceedings of the 16th International Conference on Computational Semantics, pages 40–50
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

corpus to cover 3,000 turns of speech by the end of 2025. We also release a *data statement* with the corpus to describe all relevant metadata and potential biases, following best practices for data production for NLP (Bender and Friedman, 2018; McMillan-Major et al., 2024).

- We adapt AMR to represent spontaneous speech phenomena in French, including discourse markers and backchannels.
- We provide an annotation guideline for two purposes: 1) ensure annotation consistency by clarifying aspects not specified in the original AMR annotation guideline 2) newly define how to annotate linguistic features specific to French dialogue.
- We train and evaluate an AMR parser on our dataset to showcase its practical use case. This model is further expected to serve as an annotation assitance tool.

We expect our corpus to contribute to the future development of semantic parsers for French dialogue, along with future (computational) linguistics research on French dialogical data. As noted by Wein and Opitz (2024), AMR corpora and tools are an underexplored source of data for linguistic investigation. The corpus is already getting some interest from the semantics research community, as it has been integrated in Grew (Amblard et al., 2022) and can now be explored in the tool.[3]

2 Background and Related Work

2.1 Introduction to AMR

AMR represents the meaning of texts using directed, acyclic, and rooted graphs. In an AMR graph, the nodes are 1) predicates predefined in Propbank[4] (Palmer et al., 2005), *e.g.*, `break-01` in Figure 1 or 2) English words, *e.g.*, `man` and `window` in Figure 1 or 3) AMR-specific keywords, *e.g.*, `date-entity`.

The edges of the AMR graph are labeled to indicate the relation between nodes. For example, `:ARG0` and `:ARG1` in Figure 1 respectively indicate that `man` is the agent of the predicate `break-01` and that `window` is the object of the same predicate. This predicate-argument structure is defined in Propbank.[5] An AMR graph can also be represented in textual form (see Figure 2). Although AMR is initially designed for English texts, it is also commonly used to represent non-English texts (Damonte and Cohen, 2018; Xu et al., 2021; Liu et al., 2020). In multilingual settings, two sentences in different languages that convey the same meaning (*i.e.*, sentences that are translations of each other) will share the same AMR graph.

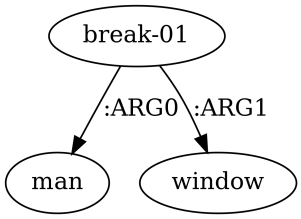


Figure 1: AMR graph for "A man breaks a window" or « Un homme a cassé la fenêtre ».

```
(b / break-01
   :ARG0 (m / man)
   :ARG1 (w / window))
```

Figure 2: AMR graph linearized in text format.

2.2 AMR Datasets

Most large-scale AMR datasets, including AMR 3.0 (Knight et al., 2020) and Massive-AMR (Regan et al., 2024), are available exclusively in English. AMR 3.0 is the most popular dataset for training and evaluating AMR parsers. It contains around 60,000 annotated examples from various sources such as news articles, blogs, and online forums. Massive-AMR, the largest manually annotated AMR dataset, consists of 84,000 utterances addressed to a virtual assistant. Most sentences in Massive-AMR are short questions or requests.

For French, a few datasets are available: *Le Petit Prince* AMR (Kang et al., 2023), Massive-AMR French (Regan et al., 2024) and ReMEDIATE (Druart, 2024). For *Le Petit Prince* AMR, the authors manually aligned the entire English dataset, *The Little Prince* AMR,[6] with the original French text. The French Massive-AMR consists of a part of Massive-AMR English (Regan et al., 2024), manually translated into French. ReMEDIATES is anno-

[3] https://semantics.grew.fr/?corpus=ding-01

[4] https://propbank.github.io/v3.4.0/frames/

[5] https://propbank.github.io/v3.4.0/frames/break.html#break.01

[6] https://github.com/flipz357/AMR-World/blob/main/data/reference_amrs/amr-bank-struct-v3.0.txt

tated semi-automatically in French using a trained annotation model. Unlike two previous datasets, ReMEDIATES is not built on pre-existing English data. In terms of corpus type, *The Little Prince* AMR is a literary piece of work. Massive-AMR consists of requests sent to virtual assistants. Finally, ReMEDIATES contains interactions between a virtual assistant and its user to make reservations. Note that ReMEDIATES uses the syntax of AMR graphs but adapts all the concepts and edge labels for Task-Oriented Dialogues (TOD).

Our work stands out from prior work in several key ways. First, we annotate spontaneous conversations between multiple speakers. Our corpus captures real-world interactions, reflecting the dynamics of spontaneous speech in French. Furthermore, *The Little Prince* AMR and Massive-AMR were initially annotated in English and then adapted to other languages through manual translation or crosslingual alignment (assuming that translated sentences should have the same semantic graph as its original sentence). This process can introduce bias, making the data potentially English-centric. We directly annotate French dialogues in AMR without relying on prior English annotations, ensuring that the semantics of French are preserved throughout the annotation process. Finally, while ReMEDIATES is annotated semi-automatically, we annotate the data manually. It is worth emphasizing that large generative language models remain unreliable for semantic annotation tasks, even for English (Ettinger et al., 2023).

2.3 AMR for Dialogues

Although standard AMR provides various semantic roles to present meanings of *texts*, several efforts have been made to extend it to capture various aspects of *dialogue*. DMR (Hu et al., 2022) and Dialogue-AMR (Bonial et al., 2020), as well as the work of Druart (2024) are among these extensions. These three approaches primarily focus on task-oriented dialogues, in which an agent requests an action to a robotic or virtual agent. Therefore, they integrate fine-grained instructions and introduce additional roles to represent, for example, illocutionary force or the speakers' intended contribution (Bonial et al., 2020).

However, these roles are not ideally suited to our corpus, which consists of spontaneous conversations among multiple speakers. While we aim to follow standard AMR conventions as closely as possible by adhering to the established annotation guidelines, the nature of our data—French dialogue—introduces linguistic phenomena specific to natural oral interaction, such as backchannels and discourse markers.

Backchannels and discourse markers convey pragmatic information in dialogue. However, standard AMR does not take this type of information into account, as specified in its annotation guidelines. Despite this, we chose to annotate the pragmatic information conveyed by backchannels and discourse markers for two main reasons. First, unlike AMR 3.0, which relies primarily on textual data, our corpus consists of dialogues rich in pragmatic content. We believe that annotating this information provides a valuable resource for the study of French dialogue. Furthermore, the additional roles we propose can be easily removed, ensuring compatibility with AMR 3.0.

Second, although the AMR annotation guideline states that pragmatic information is not included, in practice, AMR incorporates some pragmatic elements. For example, the choice of the root node in AMR often depends on the primary focus of the sentence, reflecting pragmatic information. In addition, some predicates (*e.g.,* `know-05` and `see-03`) are used for their discourse functions (*e.g.,* as in "you know" and "you see."), which are also closely related to pragmatics. Thus, adding pragmatic elements to our annotations is not entirely incompatible with standard AMR practices. To account for this pragmatic information, we introduce new roles, which are detailed in Section 5.

3 The DinG Corpus

We annotate the DinG corpus[7] (Boritchev and Amblard, 2022), a collection of manually transcribed multi-party dialogues among French-speaking players of the board game Catan.[8] Catan is a strategic board game centered on resource management and exchange. Thus, players often negotiate resource exchanges with each other, and their actual interactions are recorded in the corpus. We select this corpus for two main reasons.

First, DinG is available under a free license.[9] As

[7] https://gitlab.inria.fr/semagramme-public-projects/resources/ding/

[8] We refer readers to the website https://www.catan.com/ for more information on the game.

[9] The *Attribution ShareAlike Creative Commons* (CC BY-SA 4.0) license.

Number of utterances (non-empty)	1,667
Number of tokens covered	17,887
Number of speakers	9

Table 1: Basic statistics on our data.

our goal is to make our data public, selecting open data is a crucial requirement. Second, DinG consists of natural dialogues among speakers. Since the environment is not controlled by the data collectors and the players are free to interact during the game, this dataset captures a natural conversational flow and includes a wide variety of dialogic phenomena. As such, its semantic annotations will serve as an ideal testbed for evaluating pre-trained language models on spontaneous speech transcriptions.

4 `ding-01`

In this section, we present some statistics on the corpus, the annotation process, and the data quality assessed by inter-annotator agreement.

The annotation was carried out over a six-months period, during which approximately 1,830 (see Table 1 for other statistics) turns of speech were annotated using AMR.[10] Among these 1,830 turn takings, some examples only consist of non-annotable words, *e.g., [toux]* (cough), *[rire]* (laugh). The number of utterances (non-empty) in Table 1 excludes these non-annotable examples.

Among these examples, there are 459 discourse markers and 36 instances of *backchannel*. The corpus was primarily annotated by the first author of this article using the *metAMoRphosED* annotation tool (Heinecke, 2023, see Figure 3). Approximately 15% of the examples in the entire corpus were validated by two other annotators, who are co-authors of this article. Specifically, the lead annotator and the two annotators met regularly throughout the annotation process (once a week or every two weeks) to check the validity of the examples one by one and record any difficulties encountered. In case of disagreement among the three annotators, the example was corrected or modified during the discussion.

We encountered several challenges during the annotation process. One example concerned the word '*donc*' (so), which appears frequently in DinG. In most cases, it functions more as a discourse marker (used to start a speech turn or as a filler word) than as a causal connector. However, its usage was often ambiguous, and both interpretations could be valid depending on the context. To reduce ambiguity and improve consistency between annotations, we established the following rule: systematically annotate '*donc*' as a discourse marker, provided that its removal does not change the meaning of the sentence. Our method for addressing other similar challenges by defining clear directions is detailed in our annotation guidelines. Furthermore, when faced with complex cases, or cases where multiple annotation choices were correct, we referred to existing AMR 3.0 data in English to choose the most plausible annotation. These examples contain comments with references to the AMR 3.0 sentences that justify these choices.

To assess the quality of the annotations, 160 examples from our corpus were annotated by two annotators. The agreement score was measured using the SMATCH (Cai and Knight, 2013) score. SMATCH is an evaluation metric for AMR calculated by counting the number of triplets (node, labeled edge, node) in common. We obtained a score of 71.6. For comparison, Banarescu et al. (2013) reports inter-annotator agreement scores ranging from 71 to 83, depending on the data source and the annotators' level of expertise.

After this evaluation, we performed an annotation conflict resolution step to produce our final *gold* corpus. All three authors jointly reviewed these 160 annotation examples. In cases of disagreement, the group resolved conflicts by choosing one of the existing annotations or agreeing on a new alternative.

Common conflicts involved edge labels such as `:ARG0`, `:ARG1`, and `:ARG2`, typically resulting from annotation mistakes that were straightforward to correct once identified. Another recurring issue concerned the selection of synonymous PropBank concepts. For instance, `own-01` and `possess-01` convey the same meaning and share the same two semantic roles (`:ARG0` for the owner and `:ARG1` for the owned item). In the English AMR data, the choice between these concepts is guided by the specific lexical item used in the sentence. We used these cases of conflict to refine our annotation guidelines, ensuring a consistent selection between such synonymous concepts.

[10] We followed the original turn-taking divisions as defined in the DinG corpus.

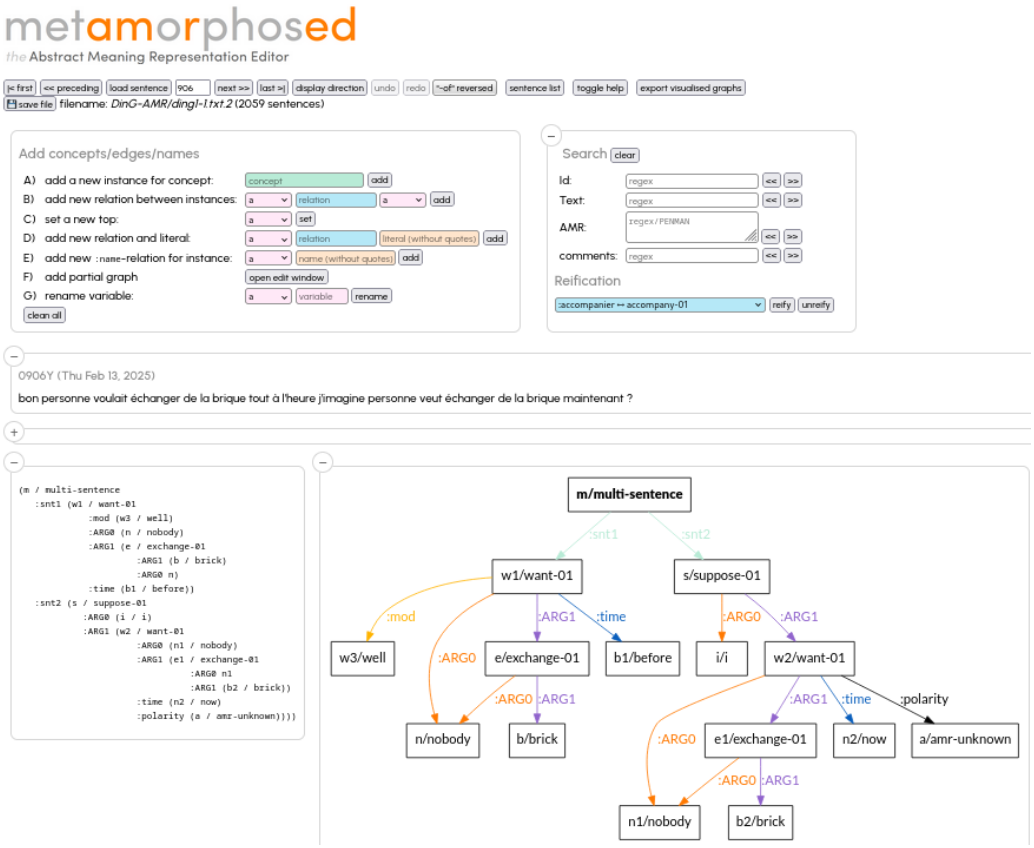


Figure 3: Screenshot illustrating the annotation process with metAMoRphosED.

5 AMR Adapted for DinG

While adhering as closely as possible to standard AMR, we introduce some extensions to better capture the specific features of spontaneous French speech. Some of these key features are outlined below. In addition, we annotate inter-instance coreference, which is an addition that sets our corpus apart from AMR 3.0. We also adapt the standard AMR concept of *focus* to represent focalization strategies in spoken French. Further details on these extensions are provided in our annotation guideline.

For `ding-01` use cases requiring compatibility with the English AMR 3.0 corpus, these extensions are designed to be easily removable.

5.1 Discourse Markers

Discourse markers are short words or phrases used by speakers to structure their discourse, for example, *donc* (so), *et* (and). They are used to begin an utterance, or can serve as fillers in the middle of an utterance or during a hesitation. We introduce a new role, `:discourse-marker`, to annotate them (see Figure 4). This role can also be reified with the concept `be-discourse-marker-91`.

```
#::id 0780B
(p / put-01
    :ARG0 (y / you)
    :ARG1 (r / road)
    :mode imperative
    :ARG2 (h / here)
    :polarity -
    :discourse-marker "donc"
```

Figure 4: « **Donc** mets pas ta route ici » (So don't put your road here).[11]

5.2 Backchannels

Backchannels refer to short interjections made by a listener while another person is speaking (e.g., *hum*, *mmh-mmh*) to signal attention to the conversation. We annotate them using a new relation `:back-channel`, which can be reified with the concept `be-back-channel-91`. Figure 6 is an annotation of backchannel to a previous utterance (Figure 5).

[11] `#::id` specifies the identifier of the example in our corpus. The identifier is composed of a number (i.e., `0780`) and the letter (i.e., `B`) that denotes a speaker.

```
#::id 0851B
(p / possible-01
   :ARG1 (e / exchange-01
      :ARG1 (t / thingy))
   :ARG1-of (r / request-confirmation-91)
   :discourse-marker "du coup"
   :time (n / now))
```

Figure 5: « du coup là on peut échanger des trucs c'est ça ? » (So now we can exchange thingies, right?).

```
#::id 0852Y
(b / be-back-channel-91
   :ARG2 "hum")
```

Figure 6: « hum » (hmm).

5.3 Inter-Instance Coreference

Since the DinG corpus captures interactions between players throughout the game, coreference can span multiple utterances or instances. To ensure a complete representation of meaning, we annotate multi-instance coreferences by marking antecedents that appear in different utterances. For example, the node `s0080b_s_stone` in Figure 8 indicates that its antecedent comes from the example identified by the ID `0080b` in Figure 7 and the concept `s / stone` associated with that example.

```
# ::id 0080B
(w / want-01
   :ARG0 (y / you)
   :ARG1 (s / stone)
   :polarity (a / amr-unknown))
```

Figure 7: « Tu veux de la **pierre** ? » (You want stone?)

```
# ::id 0082B
(e / exchange-01
   :ARG0 (I / I)
   :ARG2 (y / you)
   :ARG1 (s / sheep
      :quant 3)
   :ARG3 (s1 / s0080B_s_stone))
```

Figure 8: « Je te **l**'échange contre 3 moutons » (I trade you 3 sheep for it).

5.4 Inter-Instance Verb Ellipsis

Speakers often omit verbs when the meaning remains clear without them (verb ellipsis). When this occurs across different instances (inter-instance level), the omitted verb is mentioned in a previous utterance, and may be spoken by another speaker. We annotate such ellipses similarly to inter-instance coreference, by referencing the utterance ID of the original verb (see Figure 9 and 10).

```
# ::id 0061R
(a / and
   :op2 (p / possible-01
      :ARG1 (p1 / put-01
             :ARG0 (w / we)
             :ARG1 (c / settlement)
             :ARG2 (i / intersect-01)
             :mod (o / only))))
```

Figure 9: « On peut poser les colonies que sur les intersections. » (We can put the settlements only on intersections).

```
# ::id 0062Y
(s / s0061R_p1_put-01
   :ARG0 (w / s0061R_w_we)
   :ARG1 (r / road)
   :ARG2 (e / edge
      :mod (o / only))
```

Figure 10: « et les routes que sur les arêtes » (and roads only on edges).

5.5 Focus Representations

In AMR, the *focus* of a sentence is indicated by a root node. We apply this principle to the annotation of cleft structure, a sentence structure commonly used in French for emphasis. The cleft structure follows the pattern « C'est [subject] qui ... » ("it's [subject] who/that..." in English) used to emphasize the [subject]. To reflect this emphasis on the subject, we select it as the root of the AMR graph. Figure 11 presents an example of a sentence with a cleft structure, accompanied by its annotation in AMR. We adopt the same strategy for cases of left dislocations with pronominal resumption, as in the example: «moi, je veux 2 blés» ("me, I want 2 grains," in English). This type of structure, very common in spoken French, is also a way of expressing focus. In this case, the concept `i` will be the root of the AMR graph.

```
#::id 0095Y
(y / you
   :ARG0-of (c / choose-01
      :ARG1 (p1 / place
         :ARG2-of (p / put-01
            :ARG1 (t / they))))
   :polarity (a / amr-unknown))
```

Figure 11: « **C'est toi qui** choisis où est-ce que tu les mets ? » (It's **you** who choose where you put them?).

5.6 Disfluencies

Disfluencies are common in spontaneous dialogues. Disfluency markers (*e.g.*, euh, eh), repetitions (*e.g.*, «franchement t'es t'es franchement» "frankly

you're you're frankly" in English) and false starts (*e.g.*, «j'ai be- j'ai pas de bois» "I nee- I don't have lumber" in English) are often observed in the DinG corpus. In standard AMR, disfluency markers are not annotated. In line with this convention, we do not annotate disfluency markers, repetitions or short false starts. However, if a false start has interpretable semantic content, we annotate it using `:reparandum` (see Figure 12) following de Marneffe et al. (2021), who employed this label to mark overridden disfluencies in syntactic annotations.

```
# ::id 0314R
(t / thing
    :value 7
    :ord (o / ordinal-entity
        :value 1)
    :ARG1-of (f / fall-01)
    :ARG1-of (h / have-degree-91
        :ARG5 (r / roll-01
            :ARG1 (d / dice))
        :ARG2 (c / common
            :reparandum (p / possible-01))
        :ARG3 (m / most))
    :discourse-marker "et"
    :discourse-marker "donc"
    :discourse-marker "hein"
    :discourse-marker "et")
```

Figure 12: « et au premier 7 qui va tomber qui est donc euh le lancé de dés **le plus possible hein le plus courant** » (and the first 7 to fall, which is the most posssible the most common dice roll).

6 Models

We train an AMR parser on the previously described data to showcase its practical use. The trained model can assist in the annotation process in our future work. Specifically, the model automatically annotates the data, which can then be manually refined by a human annotator. This semi-automatic approach is useful for scaling up data annotation.

6.1 Sequence-to-Sequence AMR Parser

Recently, sequence-to-sequence AMR parsers (Konstas et al., 2017; Bevilacqua et al., 2021; Yu and Gildea, 2022) have gained popularity due to their strong performance and methodological simplicity. These models take an input sentence and generate an AMR graph in a textual format. Training such models requires a graph linearization step, which converts the AMR graph into a single-line textual format. It also requires a post-processing step because the model may produce ill-formed outputs, for example, graphs with mismatched parentheses or disconnected components. To address this, a post-processing step is applied to correct formatting errors and reconstruct a well-formed AMR graph from its linearized representation. These steps are described in more detail in the following sections.

6.2 Experimental Setup

To train a sequence-to-sequence AMR parser, we employ a multilingual language model mBart (Liu et al., 2020). To linearize AMR graph, we traverse the graph with depth first search (DFS) in line with Bevilacqua et al. (2021). As a pre-processing step, we rename variables in AMR graphs so that variable numbering follows an order (*e.g.,* a, a2, a3$\cdots$) instead of random numbering (*e.g.,* a3, a, a2$\cdots$). In addition, we added empty space between parentheses (see Figure 13 and 14 for differences between before and after pre-preprocessing).

```
(m2 / multi-sentence
    :snt1 (e / exact)
    :snt2 (m / make-05
        :ARG2 (c1 / settlement
            :ARG1-of (b / build-01
                :ARG0 (y / you)))
        :ARG1 (p / point :quant 1))
    :snt3 (m1 / make-05
        :ARG2 (c2 / city)

        :ARG1 (p1 / point :quant 2)))
```

Figure 13: AMR graph before pre-processing.

```
( m / multi-sentence
    :snt1 ( e / exact )
    :snt2 ( m2 / make-05
        :ARG2 ( s / settlement
            :ARG1-of ( b / build-01
                :ARG0 ( y / you ) ) )
            :ARG1 ( p / point :quant 1 ) )
    :snt3 ( m3 / make-05
        :ARG2 ( c / city )

        :ARG1 ( p2 / point :quant 2 ) ) )
```

Figure 14: AMR graph after pre-processing.

We train two distinct models: one trained solely on our data (hereafter referred to as Domain-specific), and another that is first trained on a larger AMR corpus (Knight et al., 2020) and then fine-tuned on our data (hereafter referred to as Pre-trained+Domain-specific). The aim of the second model is to explore whether leveraging large-scale AMR data can facilitate learning our data, which differs in several key aspects: data types (text vs. dialogue transcripts), domain (general vs. board game-related), and semantic roles (standard AMR vs. AMR adapted for French dialogue). Note that the current large-scale AMR data is only available in English and not in our target language, French.

To obtain such data in French, we translated English AMR 3.0 into French using machine translation[12] following Damonte and Cohen (2018).

We split our data set into train, dev and test sets to respectively train the model, to select the best checkpoint, and to evaluate the model's performance on unseen data. The training and dev set respectively consists of 1,375 and 146 examples.[13] For testing, we used the subset of data that underwent a conflict resolution (see Section 4), which consists of 146 examples after filtering out examples solely consisting of non-annotable words.

The model was trained for 4,000 steps, with evaluations conducted every 50 steps on a dev set to select the best-performing checkpoint. Early stopping was applied, terminating training if the validation score did not improve over 750 consecutive steps. The learning rate was set to $3e{-5}$. Pre-trained+Domain-specific was initially pre-trained on AMR 3.0 data for up to 40,000 steps, with early stopping triggered after 7,500 steps without improvement. Following pre-training, the model was fine-tuned on our data for 4,000 steps using the same settings described above for the Domain-specific training.

6.3 Results

Figure 2 shows the results of our experiments. The findings indicate that pre-training the model on large-scale data is beneficial to learn our corpus in several ways. First, it helps to learn the correct structure of AMR graphs. For example, while the Domain-specific model produced 3 ill-formed graphs out of 146 that could not be recovered during post-processing, the Pre-trained+Domain-specific model successfully avoided such errors.

Moreover, large-scale pre-training helps the model better identify the appropriate predicates for French text. The Domain-specific model occasionally produced predicates that closely resembled the surface form of the French verb, rather than the correct PropBank predicate. For instance, it generated `poser-01` instead of `put-01` for the phrase «tu peux poser...» (you can put...), and `peux-01` instead of `capable-01` for «tu peux » (you can).

[12] https://www.deepl.com/fr/translator

[13] We filtered out examples that include only non-annotable sound *e.g.,* [*rire*] and [*toux*] - [laugh] and [cough] in English.

	SMATCH
Domain-specific	68.1
Pre-trained+Domain-specific	73.5

Table 2: SMATCH scores of the two models.

Despite these improvements, both models exhibited certain weaknesses. Some sentences in the dataset included non-annotable elements such as coughing or laughter, marked with square brackets (*e.g.,* [toux] for coughing, [rire] for laughing). These elements should not be represented in AMR graphs, but our model failed to capture the pattern and incorrectly annotated some of them (see Figures 15 and 16 for an example). Additionally, although the Pre-trained+Domain-specific model generally performed better at predicting PropBank predicates for French verbs, both models struggled with rare verbs. In such cases, they generated incorrect predicates resembling the verb's surface form—for example, `confine-01` instead of `entrust-01` for «on te confie...» (we entrust you with...).

```
(y / yes
    :mod (a / ah))
```

Figure 15: Reference graph for « ah **[pron fin de mot fricative palatele sourde]**+ oui (0.5s) +[pron]» (ah [pronounce voiceless palatal fricative]+ yes (0.5s)).

```
(m / multi-sentence
   :snt1 (a / ah)
   :snt2 (e / end-01
      :ARG1 (w / word
         :mod (f / fricative))
      :ARG2 (y / yes))
   :snt3 (a2 / and
      :op1 (y2 / yes)))
```

Figure 16: Pre-trained+Domain-specific's prediction for Figure 15.

Lastly, concerning new semantic roles added in our adaptation (`:discourse-marker` and `:back-channel`), both models showed good performance at capturing them. Among 43 discourse markers to predict, both models found around 30 discourse-markers (recall around 0.7). However, some of these discourse-markers were attached to wrong parent nodes. As for `:back-channel`, there was only one example in the test set and both models correctly predicted the `:back-channel`.

7 Conclusion and Future Work

We presented our ongoing work to annotate the DinG corpus in AMR to contribute to linguistic resources for French. To better represent the dynamics of spontaneous speech in the DinG corpus, we adapted standard AMR by introducing new semantic roles. We provide an annotation guideline detailing these adaptations, as well as a data statement containing metadata of `ding-01`.[14] To demonstrate a practical application of the dataset, we trained and evaluated an AMR parser on our data. The resulting model can also serve as an annotation assistance tool, helping to accelerate the annotation process and scale up the semantic annotation process. In our future work, we aim to expand the annotated dataset to approximately 3,000 utterances.

UMR Uniform Meaning Representation (UMR) have been introduced in Van Gysel et al. (2021) as an extension of AMR to languages other than English, with the ambition of being used to "annotate the semantic content of a text in any language". UMR is developed as AMR with additional features, notably aspect, tense, modality, along with expanded ones, such as quantification & scope, and discourse relations.

While UMR appears as a very promising representation tool, we have not yet used it for our purposes. There is no French-UMR dataset available for now, which makes evaluation difficult, especially for corpora with complex language phenomena such as DinG. We plan to participate in the development of AMR to UMR translation tools, which should result in several silver French-UMR corpora, paving the way for further meaning representation work. The additions we made to AMR in order to annotate DinG are a lighter version of some of the additional annotations needed for UMR annotation; thus our annotation guidelines could also be of use for a middle step between AMR and UMR.

Acknowledgments

We thank reviewers for their comments and suggestions. We gratefully acknowledge the support of Institut Carnot Cognition (project ANAGRAM) and of the French National Research Agency (grant ANR-23-CE23-0017-01, project SynPaX).

[14]The annotation guideline, the data statement, and the corpus are available at `https://doi.org/10.5281/zenodo.15537425`.

References

Maxime Amblard, Bruno Guillaume, Siyana Pavlova, and Guy Perrier. 2022. Graph querying for semantic annotations. In *Proceedings of the 18th Joint ACL-ISO Workshop on Interoperable Semantic Annotation within LREC2022*, pages 95–101.

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.

Emily M. Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12564–12573.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie M. Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. 2020. Dialogue-AMR: Abstract Meaning Representation for dialogue. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 684–695, Marseille, France. European Language Resources Association.

Claire Bonial, Julie Foresta, Nicholas C. Fung, Cory J. Hayes, Philip Osteen, Jacob Arkin, Benned Hedegaard, and Thomas Howard. 2023. Abstract Meaning Representation for grounded human-robot communication. In *Proceedings of the Fourth International Workshop on Designing Meaning Representations*, pages 34–44, Nancy, France. Association for Computational Linguistics.

Claire N. Bonial, Lucia Donatelli, Jessica Ervin, and Clare R. Voss. 2019. Abstract Meaning Representation for human-robot dialogue. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 236–246.

Maria Boritchev and Maxime Amblard. 2022. A multiparty dialogue ressource in French. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 814–823, Marseille, France. European Language Resources Association.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association*

for Computational Linguistics (Volume 2: Short Papers), pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Marco Damonte and Shay B. Cohen. 2018. Cross-lingual Abstract Meaning Representation parsing. 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 1146–1155, New Orleans, Louisiana. Association for Computational Linguistics.

Lucas Druart. 2024. *Vers une Compréhension Contextuelle et Structurée de la Parole Dialogique Orientée Tâche*. Theses, Université d'Avignon.

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.

Johannes Heinecke. 2023. metAMoRphosED, a graphical editor for Abstract Meaning Representation. In *Proceedings of the 19th Joint ACL-ISO Workshop on Interoperable Semantics (ISA-19)*, pages 27–32, Nancy, France. Association for Computational Linguistics.

Xiangkun Hu, Junqi Dai, Hang Yan, Yi Zhang, Qipeng Guo, Xipeng Qiu, and Zheng Zhang. 2022. Dialogue meaning representation for task-oriented dialogue systems. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 223–237, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jeongwoo Kang, Maximin Coavoux, Didier Schwab, and Cédric Lopez. 2023. Analyse sémantique AMR pour le français par transfert translingue. In *Actes de CORIA-TALN 2023. Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 2 : travaux de recherche originaux – articles courts*, pages 55–62, Paris, France. ATALA.

Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, and Nathan Schneider. 2020. Abstract meaning representation (AMR) annotation release 3.0 ldc2020t02. Philadelphia: Linguistic Data Consortium.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Vancouver, Canada. Association for Computational Linguistics.

Kexin Liao, Logan Lebanoff, and Fei Liu. 2018. Abstract Meaning Representation for multi-document summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1178–1190, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. 2015. Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1077–1086, Denver, Colorado. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Angelina McMillan-Major, Emily M. Bender, and Batya Friedman. 2024. Data statements: From technical concept to community practice. *ACM J. Responsib. Comput.*, 1(1).

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Michael Regan, Shira Wein, George Baker, and Emilio Monti. 2024. MASSIVE multilingual Abstract Meaning Representation: A dataset and baselines for hallucination detection. In *Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (*SEM 2024)*, pages 1–17, Mexico City, Mexico. Association for Computational Linguistics.

Jens EL Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, et al. 2021. Designing a uniform meaning representation for natural language processing. *KI-Künstliche Intelligenz*, 35(3):343–360.

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.

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.

Dongqin Xu, Junhui Li, Muhua Zhu, Min Zhang, and Guodong Zhou. 2021. XLPT-AMR: Cross-lingual pre-training via multi-task learning for zero-shot AMR parsing and text generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 896–907, Online. Association for Computational Linguistics.

Chen Yu and Daniel Gildea. 2022. Sequence-to-sequence AMR parsing with ancestor information. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 571–577, Dublin, Ireland. Association for Computational Linguistics.

A Graph Autoencoder Approach for Gesture Classification with Gesture AMR

Huma Jamil[1] **Ibrahim Khebour**[1] **Kenneth Lai**[2]
James Pustejovsky[2] **Nikhil Krishnaswamy**[1]
[1]Colorado State University, Fort Collins, CO USA
[2]Brandeis University, Waltham, MA USA
{huma.jamil,ibrahim.khebour,nkrishna}@colostate.edu
{klai12,jamesp}@brandeis.edu

Abstract

We present a novel graph autoencoder (GAE) architecture for classifying gestures using Gesture Abstract Meaning Representation (GAMR), a structured semantic annotation framework for gestures in collaborative tasks. We leverage the inherent graphical structure of GAMR by employing Graph Neural Networks (GNNs), specifically an Edge-aware Graph Attention Network (EdgeGAT), to learn embeddings of gesture semantic representations. Using the EGGNOG dataset, which captures diverse physical gesture forms expressing similar semantics, we evaluate our GAE on a multi-label classification task for gestural actions. Results indicate that our approach significantly outperforms naive baselines and is competitive with specialized Transformer-based models like AMRBART, despite using considerably fewer parameters and no pretraining. This work highlights the effectiveness of structured graphical representations in modeling multimodal semantics, offering a scalable and efficient approach to gesture interpretation in situated human-agent collaborative scenarios.

1 Introduction

In-person situated communication involves not just language, but non-verbal behavior like actions and, importantly, gestures. However, automated gesture interpretation is complicated by how the same gestural semantics may be represented by very different physical forms. Fig. 1 shows an instance of this: two people use entirely distinct iconic gesture shapes to denote the same concept—*block*.

This points to the need for higher levels of abstraction to adequately model the relationship between physical form and gestural meaning, particularly in collaborative dialogue. Abstract Meaning Representation (AMR; Banarescu et al. (2013)) is a popular choice in the computational semantics community for its clarity and expressiveness, and Brutti et al. (2022) and Donatelli et al. (2022) developed Gesture AMR (GAMR), an AMR formalism specifically for gesture semantics. Within GAMR, the semantics accompanying the iconic gesture *block*, irrespective of physical form, may be rendered as follows:

Figure 1: Example from the EGGNOG dataset (Wang et al., 2017) showing different gesture shapes expressing the same gesture semantics. Both are *iconic* gestures (Brutti et al., 2022) denoting blocks, articulated differently: the physical label of the left is `RH: into closed, left`; that of the right is `arms: move, up; hands: into facing, into open`.

```
(i / icon
    :ARG0 (s / signaler)
    :ARG1 (b / block)
    :ARG2 (a / actor))
```

In this paper, we observe that AMR/GAMR's natural graphical structure lends itself to graph neural network (GNN)-based approaches for automated processing, and propose a graph autoencoder (GAE) that learns mappings between gesture semantics represented in GAMR annotation and the physical forms of the associated gestures. Experiments on EGGNOG (Wang et al., 2017), a challenging audio-visual dataset, show that our approach both outperforms naive baselines, and beats or competes with strong Transformers on gesture shape prediction, despite having significantly fewer parameters and faster inference time, making our

Proceedings of the 16th International Conference on Computational Semantics, pages 51–58
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

method suitable for gesture classification in low-resource and edge environments.

2 Related Work

Early work on gesture semantics followed traditions viewing gesture as simulated action (Kendon et al., 1980; Kendon, 2004) or a general mode of reference (McNeill, 1992, 2000, 2008). Following McNeill's work, Lascarides and Stone (2006, 2009) posited a division of gestures into *deictic* and *iconic*, creating a typing system continued in GAMR (Brutti et al., 2022; Donatelli et al., 2022). Lücking et al. (2015), Pustejovsky and Krishnaswamy (2021a,b, 2022), and Krishnaswamy and Pustejovsky (2021) further developed the grammar, semantics, and pragmatics of gesture on which GAMR is based. Related coding schemes for multimodal or non-verbal behavior include Kopp et al. (2006); Allwood et al. (2007); Kipp et al. (2007); Kong et al. (2015), and Rohrer et al. (2020).

Abstract Meaning Representation (AMR; (Banarescu et al., 2013)) is well-known for abstracting away from specific syntax using rooted, directed acyclic graphs (DAGs) and for applications to diverse tasks such as translation and NLU. Graph-based learning approaches using AMR include AMR-to-sequence learning (Beck et al., 2018) and text generation (Song et al., 2018; Wang et al., 2020; Zhao et al., 2020).

Our primary experimental dataset **EGGNOG** (Wang et al., 2017), containing natural gestures elicited during a collaborative task. EGGNOG has been used to train gesture recognizers for multimodal interactive agents such as Krishnaswamy et al. (2017, 2020, 2022) and Narayana et al. (2019). Lai et al. (2024) annotated a subset of EGGNOG with gesture and speech AMR, as well as coreference relations within and across the two modalities.

3 Methodology

The EGGNOG dataset (Wang et al., 2017) contains 360 videos of pairs of participants engaged in a collaborative task. One person, the actor, is given a set of wooden blocks, while the other, the signaler, is shown an image of a block structure. The signaler uses gesture, sometimes together with speech, to instruct the actor how to build the structure. Gestures are labeled according to both a *physical description* (e.g., `RH: thumbs, up`) and the signaler's *intent* (e.g., `yes`); this work focuses on the former.

Each EGGNOG physical gesture label refers to one or more body parts, which include the head, arms, hands, and upper body. Each body part is then described with one or more *aspects*, including various types of *motions* (of body parts in space, such as `rotate` and `shake`), *relations* (of body parts to each other, such as `crossed` and `facing`), and *poses* (hand positions, such as `claw` and `point`). Finally, aspects have optional orientations: `up`, `down`, `left`, `right`, `front`, or `back`. See Fig. 1 for an example. For simplicity, we focus on the *aspects* within each label.

Lai et al. (2024) annotated 21 of the EGGNOG videos with Gesture AMR. Because this was done separately from the physical gesture labels, a single GAMR can overlap with multiple labels. We link each GAMR with each overlapping label, and, in turn, with each aspect occurring in those labels, making this a multi-label classification problem. In total, the dataset contains 319 GAMRs (167 unique), associated with 889 aspects (33 unique). We split the data into an 80:20 train/test split.

3.1 Graph Autoencoder

Our graph autoencoder (GAE) learns graph-level representations from GAMR graphs for the EGGNOG classification task. It is adopted from the EdgeGAT-based message passing framework proposed by Zhang and Ji (2021), which leverages edge-aware attention mechanisms to integrate both node and edge features. Each node in the graph is represented using a one-hot 94D feature vector, where 94 is the size of the unique node vocabulary extracted from the GAMRs in the EGGNOG dataset. Edges are typed with one of 9 possible labels and are embedded into 9D continuous vectors using a learnable embedding layer. To enable bidirectional information flow between root and leaf nodes, all graphs are made explicitly bidirectional by adding the reverse of each original edge.

The encoder consists of three stacked EdgeGAT layers. Each EdgeGAT layer performs attention-based message passing where, for a given node i and neighbor j, attention score α_{ij} is computed as

$$\alpha_{ij} = \text{LeakyReLU}\left(\mathbf{a}^T \left[\mathbf{W}\mathbf{h}_i \,\|\, \mathbf{W}\mathbf{h}_j \,\|\, \mathbf{W}_e\mathbf{e}_{ij}\right]\right),$$

where $\mathbf{h}_i$ and $\mathbf{h}_j$ are input node features, $\mathbf{e}_{ij}$ is the edge feature, and $\mathbf{W}$ and $\mathbf{W}_e$ are learnable linear projections applied to node and edge features, respectively. $\mathbf{a}^T$ is a learnable linear layer that maps the concatenated vector into a scalar attention score. Post-activation, these values are normalized using softmax to compute a weighted sum over neighbor

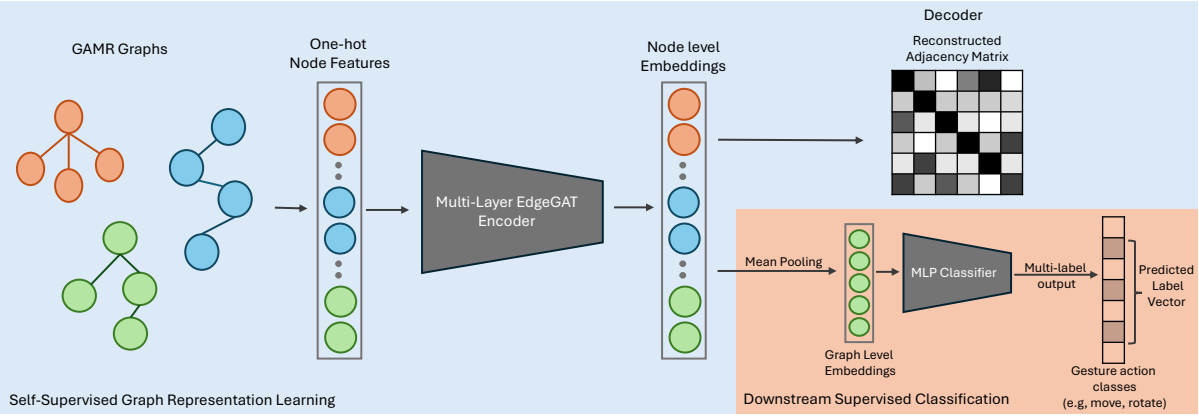


Figure 2: Graph autoencoder with EdgeGAT for self-supervised GAMR embedding, followed by MLP-based multi-label gesture classification.

embeddings. A residual connection is applied to preserve the original node features, controlled by a mixing parameter λ:

$$\mathbf{h}_i^{\text{out}} = (1-\lambda)\cdot\mathbf{h}_i + \lambda\cdot\sum_{j\in\mathcal{N}(i)}\text{softmax}_j(\alpha_{ij})\cdot\mathbf{W}\mathbf{h}_j.$$

Each EdgeGAT layer except for the last is followed by a ReLU activation. Node embeddings are then average-pooled into a fixed-dimensional graph representation $\mathbf{g} = \frac{1}{|V|}\sum_{i\in V}\mathbf{h}_i^{\text{final}}$, where V is the set of nodes and $\mathbf{h}_i^{\text{final}}$ is the embedding of node i from the last EdgeGAT layer.

We employ a multilayer perceptron (MLP) decoder to predict the presence of edges. For each edge (i, j), the decoder receives the concatenation of node embeddings $[\mathbf{z}_i \,\|\, \mathbf{z}_j]$ and outputs a scalar prediction $\hat{y}_{ij} = \sigma\left(\text{MLP}([\mathbf{z}_i \,\|\, \mathbf{z}_j])\right)$, where σ is the sigmoid activation function. The MLP consists of a 128D hidden layer, followed by ReLU, and a final linear layer projecting to a scalar.

The training objective is binary cross-entropy over observed positive and sampled negative edges:

$$\mathcal{L} = -\frac{1}{|E^+|}\sum_{(i,j)\in E^+}\log\hat{y}_{ij} - \frac{1}{|E^-|}\sum_{(i,j)\in E^-}\log(1-\hat{y}_{ij})$$

where E^+ denotes the set of observed edges and E^- is the set of randomly sampled negative edges. The model is optimized using the Adam optimizer with a learning rate of 0.001 over 100 epochs.

This GAE framework learns node and graph-level representations that capture both structural and semantic properties of the GAMR graphs. The learned graph embeddings are used for downstream classification in the EGGNOG task.

3.2 Evaluation

We evaluate the effectiveness of different vectorized GAMR representations for classifying the physical description of gestures. The EGGNOG dataset provides ELAN-annotated gesture instances along with their associated physical forms. High-level physical actions, such as *put*, *lift*, and *lean*, serve as the classification labels for this task.

The same GAMR (i.e., same graph structure) may appear multiple times across different gesture instances, each potentially annotated with a different set of physical labels. To investigate the impact of label aggregation on classification performance, we evaluate three label assignment strategies:

1. **Non-Aggregated (Instance-Level)**: Each GAMR instance is treated independently, with its own label set. This results in multiple instances of the same GAMR with potentially different labels.
2. **Majority Aggregation ($\geq$ 50%)**: For each unique GAMR, only those labels that appear in at least 50% of its instances are retained. This strategy aims to filter out noise while preserving consistent labels.
3. **Binary-Union Aggregation (Any Occurrence)**: For each unique GAMR, we include all labels that appear in **any** of its instances. This is the most inclusive strategy and ensures maximum label coverage.

All three versions result in a multi-label classification setup with 33 possible physical action aspect labels. We report results separately for each to enable informed choice of strategy for downstream task accuracy and robustness.

We compare classification performance of graph-based GAMR embeddings against several alternatives: (1) a naive baseline where GAMRs are represented using k-hot encodings of their node vocabulary, (2) embeddings of GAMRs extracted

	Instance-Level			Majority Aggregation			Binary-Union			
	P	R	F_1	P	R	F_1	P	R	F_1	# params.
k-hot	0.083	1.000	0.154	0.083	1.000	0.152	0.188	1.000	0.317	—
RoBERTa	0.475	0.479	0.477	0.602	0.599	0.601	0.772	0.833	0.802	124.1M
AMRBART	0.487	0.474	0.480	0.732	0.715	0.724	0.882	0.895	0.889	409.3M
GAE	0.490	0.447	0.468	0.731	0.648	0.687	0.834	0.836	0.835	52k

Table 1: Performance comparison of different GAMR representations across different label aggregation strategies on multi-label classification. All models use the same MLP classifier and training setup. *# params* incl. trainable and non-trainable, excl. MLP classification head.

from pretrained RoBERTa (Liu et al., 2019) using linearized AMRs as strings, and (3) GAMR embeddings from AMRBART (Bai et al., 2022) pretrained specifically on AMR parsing and generation.

For all embedding types, we use a lightweight multi-layer perceptron (MLP) classifier, consistent with common practice in unsupervised learning evaluations. The input to the classifier is the GAMR embedding vector as extracted from each method. All classifiers are trained and evaluated on the same 80:20 split described in Sec. 3.

All experiments follow the training protocol described in Sec. 3.1. This ensures that performance differences stem from the quality of the underlying GAMR representations rather than classifier capacity. We evaluate the three embedding types (GAE, AMRBART, RoBERTa), and the flat k-hot baseline, across the three aforementioned labeling strategies.

In these experiments, we use AMRBART-large-v2, which is a simpler, faster, and stronger version of AMRBART-large. This was pretrained on AMR 3.0[1], which comprises 55,635 training instances, as well as on 200,000 English sentences from English Gigaword[2]. RoBERTa experiments use RoBERTa-base.

4 Results and Discussion

Table 1 shows micro-averaged precision, recall, and F1 across all labels. The best overall performance is achieved under the **binary-union** label strategy, where a GAMR is labeled with any action that appears in at least one of its instances.

While AMRBART achieves the best F1 score overall, our GAE embeddings achieve competitive performance despite using orders of magnitude fewer parameters (Table 1, right side) and no pretraining. Notably, GAE embeddings outperform RoBERTa-based ones in both binary-union and majority aggregation settings, highlighting the benefit of incorporating relational structure over a linearized string representation. The naive k-hot baseline performs poorly all around due to its inability to encode structural context, and tends to overlabel all class, resulting in a spurious 100% recall. These results suggest that leveraging the graph structure of GAMRs provides a natural, effective and, notably, *efficient* way to learn meaningful gesture representations.

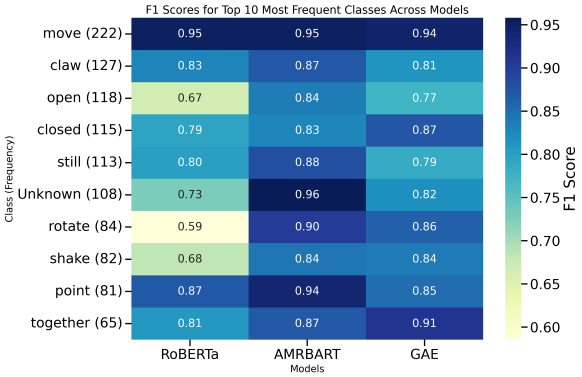


Figure 3: F1 scores for the top 10 most frequent classes across under binary-union labeling.

Table 2 shows the performance of our proposed method across the gesture types available from the EGGNOG dataset. We can observe a slight performance advantage leaning towards Iconic gestures when using instance-level labeling, which can be explained by the data imbalance toward this class as suggested by Table 3. However, under the binary-union strategy, Deixis gestures strongly outperform the other classes, this weakening the idea that the model might be biased towards any specific gesture category across labeling strategies. Instead, the strong performance of Deixis under this strategy may be attributable to the characteristic hand-shape of most deictic gestures that accompany English spoken dialogue.

Fig. 3 shows the F1 scores for the 10 most-frequently occurring physical gesture classes according to the binary-union strategy, across all

[1] https://catalog.ldc.upenn.edu/LDC2020T02

[2] https://catalog.ldc.upenn.edu/LDC2011T07

Gesture Type	Instance-Level			Majority Aggregation			Binary-Union		
	P	R	F_1	P	R	F_1	P	R	F_1
Iconic	0.525	0.489	0.506	0.692	0.537	0.605	0.624	0.653	0.638
Deixis	0.500	0.421	0.457	0.577	0.750	0.652	0.943	0.978	0.960
Emblem	0.450	0.474	0.462	0.348	0.727	0.471	0.524	0.942	0.674

Table 2: Performance comparison over different gesture types using the GAE method.

Gesture Types	Train	Test
Iconic	179	43
Deixis	54	14
Emblem	29	8

Table 3: Gesture types distribution across train and test sets.

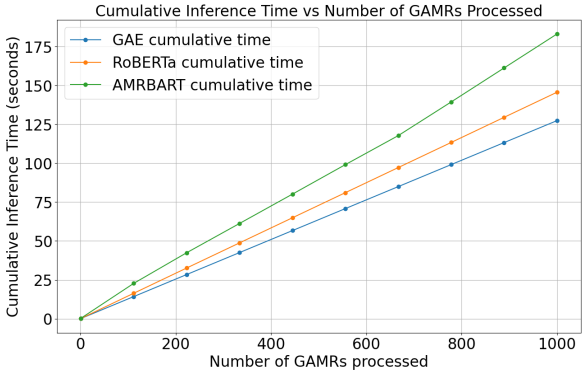


Figure 4: Cumulative inference time vs. number of GAMRs processed.

3 learnable methods. Here we see a number of classes where GAE embeddings match or exceed the performance of AMRBART embeddings, such as *closed*, *shake*, and *together*.

Finally, since the GAE has substantially fewer parameters than the competitor methods, we performed an inference-time experiment to quantify the speed advantage. Fig. 4 shows the cumulative time required to process increasing numbers of GAMRs by each method. We see that the GAE boasts a nearly 50% improvement in processing time over AMRBART despite AMRBART's extremely modest classification advantage, and that the GAE remains about 20% faster than RoBERTa at all input sizes despite outperforming it nearly globally.

5 Conclusion

We presented a novel approach to gesture classification using Gesture AMR and graph autoencoders. Our approach achieves competitive classification accuracy with SOTA Transformer approaches at significantly less computational overhead with faster inference speed. We also explored the effects of different label aggregation strategies, based on the premise that in real world data, the same semantics may have different physical forms attached to them. Our results can inform the choice of classification technique for downstream tasks that use gesture information with different requirements, such as epistemic position classification as in Khebour et al. (2024). Our efficient GAE method is suitable for real-time (e.g., VanderHoeven et al. (2025)) or GPU-less systems.

Limitations

Our method as presented (and all those tested) requires pre-annotated Gesture AMRs to be used as input, which entails additional human preparatory effort. Automating this step would entail some form of automatic AMR-graph construction for GAMR, such as sequence-to-graph transduction approaches for AMR parsing (Zhang et al., 2019) from raw dialogues and/or videos (VanderHoeven et al., 2024), potentially using text enrichment techniques such as dense paraphrasing (Tu et al., 2024).

Acknowledgments

This material is based in part upon work supported by Other Transaction award HR00112490377 from the U.S. Defense Advanced Research Projects Agency (DARPA) Friction for Accountability in Conversational Transactions (FACT) program, and by award W911NF-25-1-0096 from the U.S. Army Research Office (ARO). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

References

Jens Allwood, Loredana Cerrato, Kristiina Jokinen, Costanza Navarretta, and Patrizia Paggio. 2007. The mumin coding scheme for the annotation of feedback, turn management and sequencing phenomena. *Language Resources and Evaluation*, 41:273–287.

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for amr parsing and generation.

In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015.

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.

Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–283, Melbourne, Australia. Association for Computational Linguistics.

Richard Brutti, Lucia Donatelli, Kenneth Lai, and James Pustejovsky. 2022. Abstract Meaning Representation for gesture. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1576–1583, Marseille, France. European Language Resources Association.

Lucia Donatelli, Kenneth Lai, Richard Brutti, and James Pustejovsky. 2022. Towards situated amr: Creating a corpus of gesture amr. In *International Conference on Human-Computer Interaction*, pages 293–312. Springer.

Adam Kendon. 2004. *Gesture: Visible action as utterance*. Cambridge University Press.

Adam Kendon et al. 1980. Gesticulation and speech: Two aspects of the process of utterance. *The relationship of verbal and nonverbal communication*, 25(1980):207–227.

Ibrahim Khebour, Kenneth Lai, Mariah Bradford, Yifan Zhu, Richard Brutti, Christopher Tam, Jingxuan Tu, Benjamin Ibarra, Nathaniel Blanchard, Nikhil Krishnaswamy, and James Pustejovsky. 2024. Common ground tracking in multimodal dialogue. *arXiv preprint arXiv:2403.17284*.

Michael Kipp, Michael Neff, and Irene Albrecht. 2007. An annotation scheme for conversational gestures: how to economically capture timing and form. *Language Resources and Evaluation*, 41:325–339.

Anthony Pak-Hin Kong, Sam-Po Law, Connie Ching-Yin Kwan, Christy Lai, and Vivian Lam. 2015. A coding system with independent annotations of gesture forms and functions during verbal communication: Development of a database of speech and gesture (dosage). *Journal of nonverbal behavior*, 39:93–111.

Stefan Kopp, Brigitte Krenn, Stacy Marsella, Andrew N Marshall, Catherine Pelachaud, Hannes Pirker, Kristinn R Thórisson, and Hannes Vilhjálmsson. 2006. Towards a common framework for multimodal generation: The behavior markup language. In *Intelligent Virtual Agents: 6th International Conference, IVA 2006, Marina Del Rey, CA, USA, August 21-23, 2006. Proceedings 6*, pages 205–217. Springer.

Nikhil Krishnaswamy, Pradyumna Narayana, Rahul Bangar, Kyeongmin Rim, Dhruva Patil, David McNeely-White, Jaime Ruiz, Bruce Draper, Ross Beveridge, and James Pustejovsky. 2020. Diana's world: A situated multimodal interactive agent. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13618–13619.

Nikhil Krishnaswamy, Pradyumna Narayana, Isaac Wang, Kyeongmin Rim, Rahul Bangar, Dhruva Patil, Gururaj Mulay, Ross Beveridge, Jaime Ruiz, Bruce Draper, et al. 2017. Communicating and acting: Understanding gesture in simulation semantics. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS)—Short papers*.

Nikhil Krishnaswamy, William Pickard, Brittany Cates, Nathaniel Blanchard, and James Pustejovsky. 2022. The voxworld platform for multimodal embodied agents. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1529–1541.

Nikhil Krishnaswamy and James Pustejovsky. 2021. The role of embodiment and simulation in evaluating hci: Experiments and evaluation. In *International Conference on Human-Computer Interaction*, pages 220–232. Springer.

Kenneth Lai, Richard Brutti, Lucia Donatelli, and James Pustejovsky. 2024. Encoding gesture in multimodal dialogue: Creating a corpus of multimodal AMR. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5806–5818, Torino, Italia. ELRA and ICCL.

Alex Lascarides and Matthew Stone. 2006. *Formal semantics for iconic gesture*. Universität Potsdam.

Alex Lascarides and Matthew Stone. 2009. A formal semantic analysis of gesture. *Journal of Semantics*, 26(4):393–449.

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*.

Andy Lücking, Thies Pfeiffer, and Hannes Rieser. 2015. Pointing and reference reconsidered. *Journal of Pragmatics*, 77:56–79.

David McNeill. 1992. Hand and mind. *Advances in Visual Semiotics*, 351.

David McNeill. 2000. *Language and gesture*, volume 2. Cambridge University Press.

David McNeill. 2008. Gesture and thought.

Pradyumna Narayana, Nikhil Krishnaswamy, Isaac Wang, Rahul Bangar, Dhruva Patil, Gururaj Mulay, Kyeongmin Rim, Ross Beveridge, Jaime Ruiz, James Pustejovsky, et al. 2019. Cooperating with avatars through gesture, language and action. In *Intelligent Systems and Applications: Proceedings of the 2018 Intelligent Systems Conference (IntelliSys) Volume 1*, pages 272–293. Springer.

James Pustejovsky and Nikhil Krishnaswamy. 2021a. Embodied human computer interaction. *KI-Künstliche Intelligenz*, 35(3):307–327.

James Pustejovsky and Nikhil Krishnaswamy. 2021b. The role of embodiment and simulation in evaluating hci: theory and framework. In *International Conference on Human-Computer Interaction*, pages 288–303. Springer.

James Pustejovsky and Nikhil Krishnaswamy. 2022. Multimodal semantics for affordances and actions. In *International Conference on Human-Computer Interaction*, pages 137–160. Springer.

Patrick Louis Rohrer, Ingrid Vilà-Giménez, Júlia Florit-Pons, Núria Esteve-Gibert, Ada Ren, Stefanie Shattuck-Hufnagel, and Pilar Prieto. 2020. The multimodal multidimensional (m3d) labelling scheme for the annotation of audiovisual corpora. *Gesture and Speech in Interaction (GESPIN)*.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for AMR-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1616–1626, Melbourne, Australia. Association for Computational Linguistics.

Jingxuan Tu, Kyeongmin Rim, Bingyang Ye, Kenneth Lai, and James Pustejovsky. 2024. Dense paraphrasing for multimodal dialogue interpretation. *Frontiers in artificial intelligence*, 7:1479905.

Hannah VanderHoeven, Brady Bhalla, Ibrahim Khebour, Austin C Youngren, Videep Venkatesha, Mariah Bradford, Jack Fitzgerald, Carlos Mabrey, Jingxuan Tu, Yifan Zhu, et al. 2025. Trace: Real-time multimodal common ground tracking in situated collaborative dialogues. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)*, pages 40–50.

Hannah VanderHoeven, Nathaniel Blanchard, and Nikhil Krishnaswamy. 2024. Point target detection for multimodal communication. In *International Conference on Human-Computer Interaction*, pages 356–373. Springer.

Isaac Wang, Mohtadi Ben Fraj, Pradyumna Narayana, Dhruva Patil, Gururaj Mulay, Rahul Bangar, J Ross Beveridge, Bruce A Draper, and Jaime Ruiz. 2017. Eggnog: A continuous, multi-modal data set of naturally occurring gestures with ground truth labels. In *2017 12th Ieee international conference on automatic face & gesture recognition (fg 2017)*, pages 414–421. IEEE.

Tianming Wang, Xiaojun Wan, and Hanqi Jin. 2020. AMR-to-text generation with graph transformer. *Transactions of the Association for Computational Linguistics*, 8:19–33.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. Amr parsing as sequence-to-graph transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 80–94.

Zixuan Zhang and Heng Ji. 2021. Abstract Meaning Representation guided graph encoding and decoding for joint information extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 39–49, Online. Association for Computational Linguistics.

Yanbin Zhao, Lu Chen, Zhi Chen, Ruisheng Cao, Su Zhu, and Kai Yu. 2020. Line graph enhanced AMR-to-text generation with mix-order graph attention networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 732–741, Online. Association for Computational Linguistics.

A Additional Technical and Implementation Details

Hyperparameters Each of the 3 layers of the EdgeGAT network consists of 94 hidden units. The value of the mixing parameter λ was set to 0.5. The MLP classifier consists of 3 hidden layers (256, 128, and 64 units, respectively). Batch normalization and ReLU activation are used after each of the first three linear layers, followed by a dropout layer with probability 0.3. The activation function used throughout is ReLU.

MLP Decoder An inner product decoder models only a simple, fixed linear similarity between node embeddings. That is, it only predicts that an edge exists between two nodes if node vectors are aligned (high inner product), providing a rigid notion of connectivity. By contrast, an MLP provides a learnable decoder which can learn complex, nonlinear relationships to explain the presence and absence of edges, and hence can more reliably capture asymmetric relationships. When comparing inner product with MLP approaches during development, we used AUROC on the task of reconstructing the node adjacency matrix as a guiding metric. An inner product decoder achieved a top AUROC of 93, which increased to 99 with the MLP decoder.

Sampling Strategy Random sampling was used for sampling negative edges for training (Sec. 3.1). For each batch, we sampled node pairs that are not connected in the input graph to serve as negative edges. The sampling is uniform and done on the fly during training and evaluation. We use the `negative_sampling` utility by `torch_geometric`, which makes sure that sampled edges do not overlap with positive edges.

Hardware and Software All classification experiments were performed on an AMD Ryzen Threadripper 3960X 3.8 GHz system with 96 GB RAM running Linux 5.15.0-130-generic x86_64 (Ubuntu-based kernel).

The inference time experiment shown in Fig. 4 was performed on an Intel Xeon Gold 5520+ with 256 GB RAM and Ubuntu 24.04.2 LTS.

PyTorch 2.4.0 was used.

Retrieval-Augmented Semantic Parsing: Improving Generalization with Lexical Knowledge

Xiao Zhang
University of Groningen
xiao.zhang@rug.nl

Qianru Meng
Leiden University
q.r.meng@liacs.leidenuniv.nl

Johan Bos
University of Groningen
johan.bos@rug.nl

Abstract

Open-domain semantic parsing remains a challenging task, as neural models often rely on heuristics and struggle to handle unseen concepts. In this paper, we investigate the potential of large language models (LLMs) for this task and introduce Retrieval-Augmented Semantic Parsing (RASP), a simple yet effective approach that integrates external lexical knowledge into the parsing process. Our experiments not only show that LLMs outperform previous encoder-decoder baselines for semantic parsing, but that RASP further enhances their ability to predict unseen concepts, nearly doubling the performance of previous models on out-of-distribution concepts. These findings highlight the promise of leveraging large language models and retrieval mechanisms for robust and open-domain semantic parsing.

1 Introduction

Open-domain semantic parsing involves mapping natural language text to formal meaning representations, capturing the concepts, relations between them, and the contexts in which they appear (Oepen and Lønning, 2006; Hajič et al., 2012; Banarescu et al., 2013; Bos et al., 2017; Martínez Lorenzo et al., 2022). Such meaning representations are applied in many downstream applications—ranging from database querying to embodied question answering—where parsers must handle a vast array of concepts that may not appear in the training data. While neural encoder-decoder architectures have shown impressive performance in semantic parsing tasks, their reliance on training distributions constrains their ability to generalize, especially to out-of-distribution (OOD) concepts.

Most existing semantic parsers struggle to interpret the symbols, such as rare senses, often defaulting the unseen words to the most frequent meaning encountered during training. As a result, they fail to adapt to novel linguistic phenomena and remain limited to fixed patterns. Recent work (Zhang et al., 2025) have attempted to mitigate these limitations by encoding concept representations symbolically, forcing models to learn underlying structural knowledge from resources like WordNet (Fellbaum, 1998). However, these approaches require substantial preprocessing and intricate encodings that can be difficult for models to fully exploit.

In our work, instead, we explore the potential of large language models, powerful decoder-only architectures with strong in-context learning capabilities and extensive pretraining, to enhance the ability of semantic parsers to generalize. We pose two central research questions:

- **Do large language models outperform traditional encoder-decoder architectures in semantic parsing?** Decoder-only architectures are known to be more scalable and to internalize broader knowledge, potentially leading to stronger generalization and learning abilities. Assessing their performance in semantic parsing tasks can help reveal the architectural advantages of these decoder-only models.
- **How can these large language models be leveraged to improve the generation of out-of-distribution concepts?** Beyond simple architecture comparisons, we investigate whether LLMs can be guided to handle concepts more flexibly, using their ability to interpret and integrate external information.

In Section 2 we provide background on the semantic formalism of our choice, earlier approach to semantic parsing, and the challenge of an important task, word sense disambiguation. Then we propose **Retrieval-Augmented Semantic Parsing** (RASP) in Section 3, a technique that integrates a retrieval mechanism into parsing. RASP leverages external

Proceedings of the 16th International Conference on Computational Semantics, pages 59–72
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

lexical knowledge in the input, enabling the model to dynamically access and interpret relevant concept information. By incorporating this retrieval step (Section 4), we relax the reliance on lemma-based mappings and allow the model to adapt more naturally to unseen words or senses. Our results show that this approach nearly doubles the performance on predicting OOD concepts compared to previous methods, demonstrating a substantial advancement in handling challenging open-domain data (Section 5).

2 Background and Related Work

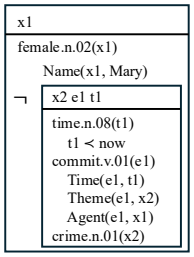
2.1 Discourse Representation Structure

Discourse Representation Theory (Kamp and Reyle, 1993, DRT) is a semantic modeling framework. The core component of DRT is the Discourse Representation Structure (DRS), a formal representation that captures the meaning of a discourse, which captures the essence of the text and covers linguistic phenomena like anaphors and temporal expressions. Unlike many other formalisms such as Abstract Meaning Representation (Banarescu et al., 2013, AMR) used for large-scale semantic annotation efforts, DRS covers logical negation, quantification, and discourse relations. Moreover, DRS is equipped with complete word sense disambiguation, and offers a language-neutral meaning representation. A Discourse Representation Structure (DRS) can be coded and visualised in various ways, which are all provided in Parallel Meaning Bank (Abzianidze et al., 2017). In formal semantics they are often pictured in a human-readable box format. The clause notation was introduced to represent DRS in a sequential format suitable for machine learning models (van Noord et al., 2018). To further simplify DRS, Bos (2023) proposed a variable-free format known as Sequence Box Notation (SBN). An example of the three different but logically equivalent formats is shown in Figure 1. Recent trends in using seq2seq models have led to a preference for sequence notation, which is also the format used in this paper.

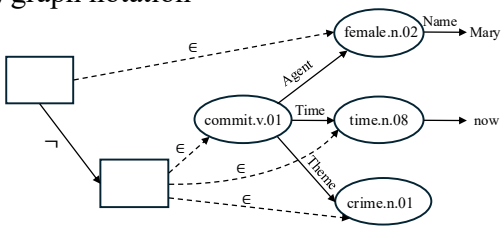
2.2 Semantic Parsing

Semantic parsing, as a traditional NLP task, remains essential in real-world applications, despite recent progress in natural language understanding shown by large language models. For instance, natural language front-end interfaces to databases require a mapping from text to structured data.

(a) box notation



(b) graph notation



(c) sequence notation

female.n.02 Name "Mary" NEGATION <1 time.n.08 TPR now
commit.v.01 Agent -2 Time -1 Theme +1 crime.n.01

Figure 1: Three formats of Discourse Representation Structure (DRS) for "Mary didn't commit a crime.": the box notation, a directed acyclic graph, and the sequence notation. These formalisms are mutually convertible without loss of information.

Speech interactions with conversational agents that act in the real world (e.g., service robots) require situation-sensitive symbol grounding. Hence, advancing the development of more robust and general semantic parsers remains crucial.

Early approaches to semantic parsing primarily relied on rule-based systems (Woods, 1973; Hendrix et al., 1977; Templeton and Burger, 1983). The advent of neural methodologies, coupled with the availability of large semantically annotated datasets (Banarescu et al., 2013; Bos et al., 2017; Abzianidze et al., 2017), marked a significant shift in semantic parsing techniques (Barzdins and Gosko, 2016; van Noord and Bos, 2017; Bevilacqua et al., 2021a). The introduction of pre-trained language models within the sequence-to-sequence framework further improved parsing performance (van Noord et al., 2018, 2020; Ozaki et al., 2020; Samuel and Straka, 2020; Shou and Lin, 2021; Bevilacqua et al., 2021a; Zhou et al., 2021; Martínez Lorenzo et al., 2022; Zhang et al., 2024; Liu, 2024a,b; Yang et al., 2024; Amin et al., 2025). Furthermore, several studies introduced more pre-training tasks specifically designed for semantic parsing (Bai et al., 2022; Wang et al., 2023a). With the rise of large language models, there has been considerable discussion about leveraging these models for

semantic parsing, achieving notable results through techniques like prompting and chain-of-thought reasoning (Roy et al., 2022; Ettinger et al., 2023; Jin et al., 2024). However, there is currently no work that leverages the knowledge and understanding capabilities of large language models to address the generalization problem in semantic parsing.

2.3 Word Sense Disambiguation

The generalization problem introduced in the previous section can also be understood as word sense disambiguation (WSD) for out-of-distribution concepts, within the context of semantic parsing. For instance, consider the sentence "She had £10,000 in the bank", with the target word "bank". In traditional WSD tasks, a predefined inventory of possible senses (e.g., 1. sloping land; 2. financial institution; 3. a long ridge or pile; 4. ...) would be provided, and the WSD model would classify the word according to one of these senses (Navigli, 2009; Bevilacqua et al., 2021b).

In semantic parsing, WSD can be seen as a subtask (Zhang et al., 2025), but it is more challenging because the parsing model must generate the correct sense directly without access to an explicitly provided sense inventory. However, traditional knowledge-based WSD offers a potential solution that inspires our approach: by retrieving and presenting all possible concepts as alternatives, we can explicitly provide external information to the model, thereby enhancing its generalization capability. As a consequence, this requires the model to be able to process long contexts, making the LLMs be the preferred choice, in particular retrieval augmented generation.

2.4 Retrieval Augmented Generation

Retrieval-Augmented Generation (RAG) is a hybrid approach that combines retrieval mechanisms with generative models to enhance the quality and accuracy of text generation tasks (Zhao et al., 2024; Gao et al., 2024). In RAG, a retrieval component first identifies relevant information from a large external knowledge base or corpus, which is then used as additional context for the generative model. This method allows the model to generate more informed and contextually accurate outputs, particularly in scenarios where the input data alone may not provide sufficient information.

By integrating retrieved knowledge into the generative process, RAG effectively bridges the gap between retrieval and generative models, leading to

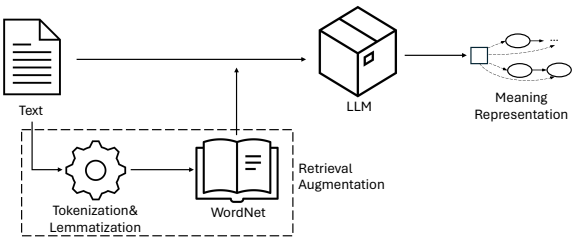


Figure 2: Global overview of RASP (Retrieval-Augmented Semantic Parsing). Both the training and testing phases adhere to this pipeline.

improved performance in tasks such as question answering (Karpukhin et al., 2020; Lewis et al., 2020; Borgeaud et al., 2021; Guu et al., 2020; Izacard and Grave, 2021; Petroni et al., 2020), commonsense reasoning (Liu et al., 2021; Wan et al., 2024) and other downstream tasks (Lewis et al., 2020; Izacard et al., 2024; Jiang et al., 2023; Guo et al., 2023; Cheng et al., 2023; Li et al., 2023). While RAG was initially employed in a wide scope of applications, its popularity can be attributed to the advent of large language models and their strong capabilities. Consequently, we will concentrate on the application of RAG in the context of LLMs.

3 Retrieval-Augmented Semantic Parsing

We propose a new method that combines retrieval-augmented generation with semantic interpretation: Retrieval-Augmented Semantic Parsing (RASP), a framework that is outlined in Figure 2. It comprises two key components: retrieval and parsing.

Different from the Dense Passage Retrieval (Karpukhin et al., 2020) method, which is commonly employed in question-answering tasks, our retrieval process is designed to be more straightforward and tailored to the needs of semantic parsing. The process begins with tokenizing and lemmatizing[1] the source text. Following these, we perform a search for relevant concept synsets in an external knowledge base, specifically WordNet. For example, in the sentence "Mary went for birdwatching. She saw a harrier, a golden eagle, and a hobby", the retrieval process would identify multiple synsets for "go", "birdwatch", "see", "harrier", "golden eagle" and "hobby", as illustrated in Table 1. Additionally, to ensure comprehensive coverage of multi-word expressions, which are critical in capturing the correct semantic meaning, we employ a hierarchical n-gram search strategy. This strategy

[1] https://www.nltk.org/api/nltk.stem.wordnet.html

Source Text	Mary went for birdwatching. She saw a harrier, a golden eagle, and a hobby.	
Concepts	golden_eagle.n.01:	large eagle of mountainous regions of ... having a golden-brown head and neck
	birdwatch.v.01:	watch and study birds in their natural habitat
	...	...
	harrier.n.01:	a persistent attacker
	harrier.n.02:	a hound that resembles a foxhound but is smaller
	harrier.n.03:	hawks that hunt over meadows and marshes and prey on small terrestrial animals ...
	...	...
	hobby.n.01	an auxiliary activity
	hobby.n.02	a child's plaything consisting of an imitation horse mounted on rockers ...
	hobby.n.03	small Old World falcon formerly trained and flown at small birds
Prompts	Normal prompt:	Text to parse: {*Source Text*}
	RASP prompt:	Considering the concepts with glosses: {*Concepts*}. Text to parse: {*Source Text*}
Gold DRS	female.n.02 Name "Mary" time.n.08 TPR now birdwatch.v.01 Agent -2 Time -1 ELABORATION <1 female.n.02 ANA -3 see.v.01 Experiencer -1 Time +1 Stimulus +3 time.n.08 TPR now harrier.n.03 golden_eagle.n.01 entity.n.01 Sub -2 Sub -1 Sub +1 hobby.n.03	

Table 1: An example illustrating the workflow of RASP. We omit some senses and words for the retrieved concepts to save space. The distinction between prompts for semantic parsing with and without RASP are shown in the *Prompts* row. Some examples of complete prompts can be found in Appendix A.

involves sequential searches using 4-gram, 3-gram, 2-gram, and 1-gram patterns, thereby ensuring that no multi-word expressions (such as "golden eagle") are overlooked.

The parsing process for a decoder-only model[2] is guided by the probability distribution of possible output sequences given an input sequence. The model generates an output sequence by predicting each token iteratively, based on the input text and previously generated tokens, as shown in (1).

$$p_{\text{decoder-only}}(o \mid x) = \prod_{i=1}^{n} p_\theta(o_i \mid x,\, o_{1:i-1}) \tag{1}$$

Here, x is the input text, $o_{1:i-1}$ represents the sequence generated so far, and o_i is the token generated at the current step. θ refers to the model's parameters, and p denotes the likelihood of generating output sequence o given input sequence x.

To enhance this process, retrieval and generation are integrated, leveraging external knowledge to inform output generation. Mathematically, the retrieval step introduces a probability, $p(o'|x)$, which models the likelihood of retrieving relevant concepts o' based on x. This probability is combined multiplicatively with the generation probability, as shown in (3). This combination ensures both components contribute meaningfully, with retrieval acting as a filter to guide the generation process toward relevant concepts.

$$\begin{aligned} p_{\text{RASP}}(o \mid x) &= p(o' \mid x)\ p_{\text{decoder-only}}(o \mid x,\, o') \\ &= p(o' \mid x) \prod_{i=1}^{n} p_\theta(o_i \mid x,\, o',\, o_{1:i-1}) \end{aligned} \tag{2}$$

By incorporating retrieved concepts, RASP goes beyond relying solely on the input sequence and training data, adding additional context to guide generation. For example, when handling words with multiple meanings, like "hobby," retrieved synsets help the model select the correct interpretation based on glosses and context. This integration sharpens the model's focus on relevant concepts, reducing the likelihood of generating incorrect or overly broad outputs, particularly for out-of-distribution concepts.

4 Experiments

4.1 Datasets

We conduct our experiments on the Parallel Meaning Bank (PMB, version 5.1.0)[3] (Abzianidze et al., 2017; Zhang et al., 2024). We first use the gold-standard English data of the PMB to evaluate the large language models and their retrieval-augmented version under in-distribution conditions.

[2]The models we use are all in decoder-only architecture, so we omit the discussion about encoder-decoder architecture.

[3]https://pmb.let.rug.nl/releases

To further assess the models' ability to handle out-of-distribution (OOD) concepts, we adopt the challenge set proposed by Zhang et al. (2025), which is also derived from the PMB. Neural semantic parsers often default to the first sense of unknown concepts–an approach that can lead to "lucky guesses" without truly understanding new words. The challenge set, consisting of 500 sentences, is deliberately designed to eliminate this shortcut. Each sentence includes at least one concept that does not appear in the training data and does not correspond to the first sense in the ontology. In total, the challenge set contains 410 unknown nouns, 128 verbs, and 65 modifiers (adjectives and adverbs). By evaluating on this set, we measure the true generalization capability of the models, testing whether they can correctly interpret novel concepts rather than relying on heuristic assignment.

Train	Dev	Standard	Challenge
9,560	1,195	1,195	500

Table 2: Dataset statistics for PMB 5.1.0, i.e., number of meaning representations for train, development and two test sets: standard and challenge.

4.2 Experiment Settings

It is crucial to note that large language models, when used in zero-shot or few-shot scenarios, tend to perform poorly on the highly complex graph structures inherent in formal meaning representations such as DRS. Prior work (Ettinger et al., 2023; Zhang et al., 2025) demonstrates that without fine-tuning, LLMs struggle to match the performance of models specifically optimized for these tasks. Therefore, in our experiments, we fine-tune all large language models.

For RASP, we explore two retrieval-enhanced approaches: (1) Train+Test Retrieval: Incorporate retrieval-derived concepts both during training and inference, thereby familiarizing the model with external lexical knowledge throughout the entire learning process. (2) Test-Only Retrieval: Use retrieval only during inference, training the model on raw DRS structures without external lexical inputs. Our experiments show that the first approach consistently yields better performance. Thus, we focus our primary analysis on the first approach and provide results for the second approach in Appendix C.

Due to computational constraints, we select open-sourced LLMs with model sizes under 10B parameters, including phi3-4B, Mistral-7B, LLaMa3.1-3B, LLaMa3.2-8B, Gemma2-2B, Gemma2-9B, Qwen2.5-3B, and Qwen2.5-7B. These models strike a balance between state-of-the-art language understanding and manageable resource requirements. For fine-tuning, we employ Low-Rank Adaptation (Hu et al., 2021, LoRA), a parameter-efficient technique that introduces trainable low-rank matrices into the model's layers, greatly reducing computational overhead.

We compare our results against several strong baselines, including BART, T5, byT5, TAX-parser (Zhang et al., 2024), and AMS-Parser (Yang et al., 2024), all of which were previously fine-tuned on PMB data. We exclude work conducted on earlier versions of PMB or using silver data. Additionally, we do not apply retrieval augmentation to these baseline models due to input length constraints, which limit their ability to incorporate external lexical sources efficiently.

We trained each model for 10 epochs, using a learning rate of 10^{-4}, and fp16 precision. More information on the hyperparameters is provided in Appendix B.

4.3 Evaluation Metrics

We used SMATCH and its variants to evaluate the performance of the models. SMATCH (Cai and Knight, 2013), referred to as Hard-SMatch, strictly matches concepts, where any discrepancy results in a non-match. In contrast, its variant, Soft-SMatch (Opitz et al., 2020), considers concept similarity when matching. Instead of adopting the approach of using word-embedding similarity, we applied the Wu-Palmer similarity (Wu and Palmer, 1994), as introduced by Zhang et al. (2024). Wu-Palmer similarity provides a precise measure of semantic similarity between concepts based on their positions within the WordNet taxonomy. Unlike embedding-based methods, it does not rely on external training and easily adapts to changes in WordNet's structure or content. The calculation is:

$$\text{WuP} = 2 * \frac{depth(\text{LCS}(s_1, s_2))}{depth(s_1) + depth(s_2)} \quad (3)$$

where s is the concept, LCS refers to the Least Common Subsumer of these concepts, and depth denotes the distance from the concept to the root of the taxonomy.

Model	Size	Input	Graph-level			Node-level
			Hard-SMatch↑	Soft-SMatch↑	IFR↓	F score↑
BART-large	400M	Normal	79.54	82.81	3.92	75.40
T5-large	770M	Normal	84.27	86.44	6.41	79.88
byT5-large	580M	Normal	87.41	89.43	4.78	84.75
AMS-Parser	–	Normal	87.08	89.15	0.00	85.00
TAX-Parser	580M	Normal	86.65	91.80	2.34	80.12
Phi3	4B	Normal	85.74	87.92	4.94 (59)	81.60
		RASP	85.96 (+0.3%)	88.13 (+0.2%)	4.80 (57)	83.33 (+2.1%)
Mistral	7B	Normal	89.95	92.48	2.00 (24)	83.90
		RASP	90.95 (+1.1%)	93.33 (+0.9%)	1.58 (19)	85.00 (+1.3%)
Qwen2.5	3B	Normal	86.50	88.64	4.69 (56)	82.60
		RASP	88.70 (+2.5%)	90.74 (+2.4%)	3.01 (36)	83.90 (+1.6%)
	7B	Normal	89.88	91.83	2.51 (30)	84.50
		RASP	89.93 (+0.1%)	91.87 (+0.1%)	2.51 (30)	85.50 (+1.2%)
LLama3	3B	Normal	87.30	90.01	3.34 (40)	81.50
		RASP	87.76 (+0.5%)	90.51 (+0.6%)	3.01 (36)	82.30 (+1.0%)
	8B	Normal	89.92	92.46	2.09 (25)	83.90
		RASP	90.65 (+0.8%)	93.10 (+0.7%)	**1.50** (18)	84.72 (+1.0%)
Gemma2	2B	Normal	89.20	91.08	3.01 (36)	84.20
		RASP	89.30 (+0.1%)	91.23 (+0.2%)	3.10 (37)	85.58 (+1.6%)
	9B	Normal	90.72	93.15	1.67 (20)	84.67
		RASP	**91.37** (+0.7%)	**93.65** (+0.5%)	1.58 (19)	**86.11** (+1.7%)

Table 3: Performance of baseline models, large language models (Normal) and their retrieval-augmented variants (RASP) on standard test, with percentage changes in parentheses. Size is the number of model's parameters (B: billion). IFR is Ill-Formed Rate and the number of ill-formed prediction are in parentheses. Note: AMS-Parser (Yang et al., 2024) performs well for IFR for it is a compositional neuro-symbolic system. TAX-Parser (Zhang et al., 2025) is a neuro-symbolic system, trained with a novel encoded meaning representation.

For the fine-grained evaluation on the challenge set, we applied the metric proposed by Wang et al. (2023b), focusing specifically on concept-node matching scores. When evaluating the results on the challenge set, we directly calculated the Wu-Palmer similarity between the target concepts and the corresponding model-generated results.

5 Results

5.1 Semantic Parsing on Standard Test

Table 3 shows that large language models consistently surpass earlier encoder-decoder baselines, providing direct evidence for our first research question. While BART, T5, and byT5 achieve Hard-SMatch scores up to 87.41, several LLM-based models (e.g., Mistral-7B, Gemma2-9B) exceed 90.0 on the standard test set. This improvement is substantial, with the strongest baseline LLM reaches 90.72 on Hard-SMatch, outperforming the best encoder-decoder model (byT5) by a margin of 3.3 points.

These higher scores are also reflected in Soft-SMatch and node-level F-scores, indicating that LLM-based models not only produce more structurally accurate meaning representations but also more reliably identify concept nodes. Additionally, Ill-Formed Rate (IFR) reductions suggest that these models generate fewer ill-structured outputs. In summary, these improvements highlight that large language models outperform previous encoder-decoder models.

Beyond confirming the advantages of LLMs, we also examine the impact of retrieval augmentation (RASP) on standard test results. Although the largest gains from retrieval are observed on the challenge set (as discussed in Section 5.2), even here on the in-distribution standard test, RASP provides consistent performance improvements. Most LLMs show an increase of about 0.3% to 2.5% in Hard-SMatch and Soft-SMatch scores when using RASP. Furthermore, the Ill-Formed Rate (IFR) tends to decrease, and the node-level F-score improves by approximately 1.0% to 2.1%. These node-level gains suggest that RASP's improvements stem largely from more accurate concept prediction. While these enhancements are moderate in the standard test scenario, they indicate that retrieval can enhance the model's understanding of concept-level semantics.

Model	Input	Noun	Verb	Modifiers	Overall
BART-large	Normal	26.11	37.34	46.88	30.95
T5-large	Normal	25.48	35.21	41.28	29.45
byT5-large	Normal	27.59	39.14	44.70	32.13
TAX-Parser	Normal	42.15	31.58	43.27	39.68
phi3-4B	Normal	35.48	36.91	46.97	37.91
	RASP	62.03 (+74.8%)	46.32 (+25.5%)	63.63 (+35.5%)	58.28 (+53.7%)
Mistral-7B	Normal	38.02	40.61	50.00	39.87
	RASP	72.03 (+89.5%)	59.27 (+46.0%)	67.42 (+34.8%)	68.44 (+71.7%)
Qwen2.5-7B	Normal	38.51	37.52	46.97	39.12
	RASP	66.77 (+73.4%)	56.95 (+51.8%)	64.39 (+37.1%)	64.12 (+63.8%)
LLama3.2-8B	Normal	37.06	34.79	47.73	37.59
	RASP	72.28 (+95.1%)	61.62 (+77.1%)	66.67 (+39.7%)	69.86 (+85.9%)
Gemma2-9B	Normal	39.68	45.01	55.30	42.54
	RASP	**73.93** (+86.3%)	**62.31** (+36.5%)	**69.70** (+26.0%)	**70.41** (+65.6%)

Table 4: Wu-Palmer similarities between unknown concepts and generated concepts across four parts of speech. For the sake of clarity, we exclude the smaller version of the same model.

5.2 Performance on the Challenge Set

Table 4 provides the results on the challenge set, designed specifically to test the models' ability to predict out-of-distribution (OOD) concepts. Here, we report Wu-Palmer similarities for unknown nouns, verbs, and modifiers (adjectives and adverbs). We calculate the Wu-Palmer similarities between the target concepts (out-of-distribution concepts) and the generated concepts (see examples in Table 5).

Among the baselines, TAX-Parser (Zhang et al., 2025) stands out, achieving an overall similarity score of 39.68. However, some Normal (non-RASP) large language models already exceed this performance on the challenge set. For example, Gemma2-9B (Normal) obtains an overall score of 42.54, indicating that LLMs can yield improvements, even without retrieval augmentation. When retrieval augmentation (RASP) is introduced, these large language models show substantial additional gains. For example, Gemma2-9B (RASP) achieves an overall similarity score of 70.41, compared to the best baseline's 39.68—an increase of over 30 absolute points. These gains are particularly remarkable for noun concepts, with relative improvements of approximately 70% to 95%. Verbs show increases between about 25% and 77%, and modifiers improve by roughly 26% to 43%.

These results directly support our second research question regarding improving out-of-distribution generalization. While model scaling alone can yield moderate improvements, the integration of external lexical knowledge through retrieval allows LLMs to select more accurate concepts in OOD scenarios. In effect, RASP helps the models "look up" relevant information, enhancing their concept selection and producing more semantically appropriate results. In this case, retrieval-augmented LLMs not only outperform strong baselines like TAX-Parser but also set the state-of-the-art for OOD semantic parsing performance.

5.3 Error Analysis on the Challenge Set

We selected a subset of the challenge set and manually checked how the best performing model—Gemma2-9B (Normal) and Gemma2-9B (RASP)—handle the out-of-distribution concepts.

We picked 22 instances, comprising 11 completely perfect predictions (WuP=1.00) and 11 imperfect predictions (WuP<1.00) made by RASP, as presented in Table 5. With respect to the perfect predictions, it is evident that the retrieval significantly enhances the model's ability to interpret most out-of-distribution concepts. For instance, in the text about birdwatching, the word "hobby" clearly refers to a species of bird. The model without RAG defaults to the most frequent sense number, predicts hobby.n.01 (an auxiliary activity). In contrast, retrieval provides the glosses of each sense related to the noun "hobby" and leads the model to pick hobby.n.03 (a falcon), by explicit lexical connections between "falcon" in the gloss of hobby.n.03 and the context provided by "birdwatching".

However, RASP makes imperfect predictions sometimes. We identified three possible causes: (a) similar glosses between WordNet concepts; (b) insufficient textual context; and (c) limitations in

Input Text	Gold	Normal	RASP
He bought the painting for a **song** on a flea market.	song.n.05	n.03 (0.22)	n.05 (1.00)
The detective **planted** a bug in the suspect's office to gather evidence.	plant.v.05	v.02 (0.22)	v.05 (1.00)
Scientist examines the insect's **antennae**.	antenna.n.03	n.01 (0.24)	n.03 (1.00)
I've seen a short **extract** from the film.	extract.n.02	n.01 (0.25)	n.02 (1.00)
She prepared a three **course** meal.	course.n.07	n.03 (0.27)	n.07 (1.00)
The music student practiced the **fugue**.	fugue.n.03	n.02 (0.28)	n.03 (1.00)
Johanna went birdwatching. She saw a harrier, a kite, and a **hobby**.	hobby.n.03	n.02 (0.38)	n.03 (1.00)
A harrier is a **muscular** dog with a hard coat.	muscular.a.02	a.01 (0.50)	a.02 (1.00)
The hiker spotted an **adder** sunbathing on a rock.	adder.n.03	n.01 (0.50)	n.03 (1.00)
A tiny **wren** was hiding in the shrubs.	wren.n.02	n.01 (0.55)	n.02 (1.00)
Hungarian is a challenging language with 18 cases.	hungarian.n.02	n.01 (0.11)	n.02 (1.00)
The moon is **waxing**.	wax.v.03	v.03 (1.00)	v.02 (0.75)
The function **ordered** the strings alphabetically.	order.v.05	v.02 (0.17)	v.06 (0.75)
The elephant's trunk is an **extended** nose.	extended.a.03	a.01 (0.50)	a.01 (0.50)
A **tripper** helps control the flow of materials on a conveyor.	tripper.n.04	n.02 (0.40)	n.02 (0.40)
We saw a **kite** gliding in the sky during the walking.	kite.n.04	n.03 (0.40)	n.03 (0.40)
The elegant **pen** glided gracefully across the tranquil lake.	pen.n.05	n.01 (0.36)	n.01 (0.36)
The immature sparrows are **feathering** already.	feather.v.05	v.03 (0.20)	v.02 (0.29)
The visitors can **observe** various species of ray in the aquarium.	observe.v.02	v.01 (0.25)	v.01 (0.25)
She **hobbled** the horse. It freaked out.	hobble.v.03	v.01 (0.18)	v.02 (0.18)
The gardener noticed the **growth** on the rose after the rain.	growth.n.04	n.01 (0.18)	n.01 (0.18)
The surge **alarmed** the town's residents.	alarm.v.02	v.01 (0.15)	v.01 (0.15)

Table 5: Twenty instances of the challenge set with content words with out-of-distribution concepts in bold face, and the concepts generated by the Gemma2-9B (Normal) and retrieval-augmented Gemma2-9B (RASP). The scores in brackets are the Wu-Palmer Similarity between the predicted concept and gold concept.

the model's linguistic coverage.

The verb "alarm" in Table 5 is an instance of the similarity problem. The challenge arises because some of its senses have similar glosses, such as alarm.v.01 (fill with apprehension or alarm) and alarm.v.02 (warn or arouse to a sense of danger). Similar issues occur with the verbs "wax", "order", "observe" and "hobble". Although glosses were carefully crafted by lexicographers, they don't always show a clear difference in meaning (Mihalcea and Moldovan, 2001; Navigli, 2006).

In cases of insufficient textual context, such as with the noun "kite" in the sentence "We saw a kite gliding in the sky", the sense annotators chose kite.n.04 (a bird of prey). However, kite.n.03 (a plaything) could perhaps also be appropriate given the limited context provided by this sentence. Similar issues can be raised in the sentences with the noun "tripper" and the verb "feathering".

The third cause can be attributed to the model's linguistic coverage. A case in point is "pen": the meanings of pen.n.01 (a writing implement) and pen.n.05 (a female swan) are quite different, but the latter is the correct one in the text "Jane saw two swans. The elegant pen glided gracefully across the tranquil lake". However, the model fails to distinguish them, likely because "pen" is rarely used to refer to "swan" in available corpora. As a result, the models may not have encountered this sense during training, making it challenging for them to predict a meaning they have not been exposed to. In sum, while retrieval drastically improves concept prediction, there are still some difficulties that can pose challenges for the models.

6 Conclusion

This paper demonstrates that LLMs, even without retrieval augmentation, outperform previous encoder-decoder approaches in semantic parsing for Discourse Representation Structures, thereby answering our first research question in the affirmative, setting a new state of the art. We also show that our proposed Retrieval-Augmented Semantic Parsing (RASP) framework, which integrates external lexical knowledge, further enhances the performance of LLMs. Notably, RASP nearly doubles the accuracy on out-of-distribution concepts, which answers our second research question and confirms robust generalization ability of RASP in open-domain scenarios. Our experiments show that by simply appending relevant information to the model input, the RASP approach offers a practical and intuitive approach that can be easily applied to other meaning representations used in natural language processing, such as AMR (Banarescu et al., 2013) and BMR (Martínez Lorenzo et al., 2022).

7 Limitations

We think the limitations of this work mainly come from two aspects: the language models used in RASP and the retrieval source (i.e., WordNet).

The retrieval process is proven to provide more information and knowledge to the models. However, retrieval will significantly increase the input length of the model, making it (only) adoptable for the large language models with strong context understanding and long text processing capabilities. Therefore, the RASP framework cannot be directly used to improve previous parsers that rely on other methods, which is also why we only provided results of retrieval-augmented LLMs.

Another limitation is the retrieval source. Our implementation of RASP uses WordNet, so if a sense is not in WordNet, it will never be guessed. For example, "velvet scooter" (a bird) is not in WordNet, nor is Cobb salad (a dish). Hence, RASP will never make a perfect prediction for such cases. Moreover, the glosses in WordNet, even though carefully crafted by lexicographers in most cases, are sometimes concise, lacking information to separate them from other senses. This makes it difficult for the models to accurately distinguish between different meanings (see Section 5.3). For future work, the BabelNet, ConceptNet, or extended WordNet (Delmonte and Rotondi, 2012; Navigli and Ponzetto, 2012; Delmonte and Rotondi, 2015; Speer et al., 2017) can be considered as a better choice for concept in meaning representations.

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Muhammad Saad Amin, Xiao Zhang, Luca Anselma, Alessandro Mazzei, and Johan Bos. 2025. Semantic processing for urdu: corpus creation, parsing, and generation. *Language Resources and Evaluation*, pages 1–32.

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015, Dublin, Ireland. 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 *LAW@ACL*.

Guntis Barzdins and Didzis Gosko. 2016. RIGA at SemEval-2016 task 8: Impact of Smatch extensions and character-level neural translation on AMR parsing accuracy. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1143–1147, San Diego, California. Association for Computational Linguistics.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021a. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. In *AAAI Conference on Artificial Intelligence*.

Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021b. Recent trends in word sense disambiguation: A survey. In *International Joint Conference on Artificial Intelligence*, pages 4330–4338. International Joint Conference on Artificial Intelligence, Inc.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, T. W. Hennigan, Saffron Huang, Lorenzo Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and L. Sifre. 2021. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning*.

Johan Bos. 2023. The sequence notation: Catching complex meanings in simple graphs. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 195–208, Nancy, France. Association for Computational Linguistics.

Johan Bos, Valerio Basile, Kilian Evang, Noortje Venhuizen, and Johannes Bjerva. 2017. The groningen meaning bank. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, volume 2, pages 463–496. Springer.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Daixuan Cheng, Shaohan Huang, Junyu Bi, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Furu Wei, Weiwei Deng, and Qi Zhang. 2023. UPRISE:

Universal prompt retrieval for improving zero-shot evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12318–12337, Singapore. Association for Computational Linguistics.

Rodolfo Delmonte and Agata Rotondi. 2012. Treebanks of logical forms: they are useful only if consistent. In *Proceedings of SAIIMRT*, pages 21–28. LREC-Paris.

Rodolfo Delmonte and Agata Rotondi. 2015. A logical form parser for correction and consistency checking of lf resources. In *Natural Language Processing and Cognitive Science*, pages 63–82. Libreria Editrice Cafoscarina.

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.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Zhicheng Guo, Sijie Cheng, Yile Wang, Peng Li, and Yang Liu. 2023. Prompt-guided retrieval augmentation for non-knowledge-intensive tasks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10896–10912, Toronto, Canada. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Ondřej Bojar, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. 2012. Announcing Prague Czech-English Dependency Treebank 2.0. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3153–3160, Istanbul, Turkey. European Language Resources Association (ELRA).

Gary G. Hendrix, Earl D. Sacerdoti, Daniel Sagalowicz, and Jonathan Slocum. 1977. Developing a natural language interface to complex data. In *TODS*.

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

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2024. Atlas: few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24(1).

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

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.

Hans Kamp and Uwe Reyle. 1993. From discourse to logic - introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory. In *Studies in Linguistics and Philosophy*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Xinze Li, Zhenghao Liu, Chenyan Xiong, Shi Yu, Yu Gu, Zhiyuan Liu, and Ge Yu. 2023. Structure-aware language model pretraining improves dense retrieval on structured data. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11560–11574, Toronto, Canada. Association for Computational Linguistics.

Jiangming Liu. 2024a. Model-agnostic cross-lingual training for discourse representation structure parsing. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11486–11497, Torino, Italia. ELRA and ICCL.

Jiangming Liu. 2024b. Soft well-formed semantic parsing with score-based selection. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15037–15043, Torino, Italia. ELRA and ICCL.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and S Yu Philip. 2021. Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6418–6425.

Abelardo Carlos Martínez Lorenzo, Marco Maru, and Roberto Navigli. 2022. Fully-Semantic Parsing and Generation: the BabelNet Meaning Representation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1727–1741, Dublin, Ireland. Association for Computational Linguistics.

Rada Mihalcea and Dan I. Moldovan. 2001. Automatic generation of a coarse grained wordnet. In *NAACL Workshop on WordNet and Other Lexical Resources*.

Roberto Navigli. 2006. Reducing the granularity of a computational lexicon via an automatic mapping to a coarse-grained sense inventory. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, pages 841–844, Genoa, Italy. European Language Resources Association (ELRA).

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2).

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artif. Intell.*, 193:217–250.

Rik van Noord, Lasha Abzianidze, Antonio Toral, and Johan Bos. 2018. Exploring neural methods for parsing discourse representation structures. *Transactions of the Association for Computational Linguistics*, 6:619–633.

Rik van Noord and Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *Computational Linguistics in the Netherlands Journal*, 7:93–108.

Rik van Noord, Antonio Toral, and Johan Bos. 2020. Character-level representations improve DRS-based semantic parsing even in the age of BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4587–4603, Online. Association for Computational Linguistics.

Stephan Oepen and Jan Tore Lønning. 2006. Discriminant-based MRS banking. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Juri Opitz, Letitia Parcalabescu, and Anette Frank. 2020. AMR similarity metrics from principles. *Transactions of the Association for Computational Linguistics*, 8:522–538.

Hiroaki Ozaki, Gaku Morio, Yuta Koreeda, Terufumi Morishita, and Toshinori Miyoshi. 2020. Hitachi at MRP 2020: Text-to-graph-notation transducer. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 40–52, Online. Association for Computational Linguistics.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *AKBC 2020*.

Subhro Roy, Sam Thomson, Tongfei Chen, Richard Shin, Adam Pauls, Jason Eisner, Benjamin Van Durme, Microsoft Semantic Machines, and Scaled Cognition. 2022. Benchclamp: A benchmark for evaluating language models on syntactic and semantic parsing. In *Neural Information Processing Systems*.

David Samuel and Milan Straka. 2020. ÚFAL at MRP 2020: Permutation-invariant semantic parsing in PERIN. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 53–64, Online. Association for Computational Linguistics.

Ziyi Shou and Fangzhen Lin. 2021. Incorporating EDS graph for AMR parsing. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 202–211, Online. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Marjorie Templeton and John Burger. 1983. Problems in natural-language interface to DBMS with examples from EUFID. In *First Conference on Applied Natural Language Processing*, pages 3–16, Santa Monica, California, USA. Association for Computational Linguistics.

Alexander Wan, Eric Wallace, and Dan Klein. 2024. What evidence do language models find convincing? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7468–7484, Bangkok, Thailand. Association for Computational Linguistics.

Chunliu Wang, Huiyuan Lai, Malvina Nissim, and Johan Bos. 2023a. Pre-trained language-meaning models for multilingual parsing and generation. In *Findings of the Association for Computational Linguistics*, page 5586–5600. Association for Computational Linguistics (ACL).

Chunliu Wang, Xiao Zhang, and Johan Bos. 2023b. Discourse representation structure parsing for Chinese. In *Proceedings of the 4th Natural Logic Meets Machine Learning Workshop*, pages 62–74, Nancy, France. Association for Computational Linguistics.

William A Woods. 1973. Progress in natural language understanding: an application to lunar geology. In *Proceedings of the June 4-8, 1973, national computer conference and exposition*, pages 441–450.

Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, Las Cruces, New Mexico, USA. Association for Computational Linguistics.

Xiulin Yang, Jonas Groschwitz, Alexander Koller, and Johan Bos. 2024. Scope-enhanced compositional semantic parsing for DRT. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19602–19616, Miami, Florida, USA. Association for Computational Linguistics.

Xiao Zhang, Gosse Bouma, and Johan Bos. 2025. Neural semantic parsing with extremely rich symbolic meaning representations. *Computational Linguistics*, 51(1):235–274.

Xiao Zhang, Chunliu Wang, Rik van Noord, and Johan Bos. 2024. Gaining more insight into neural semantic parsing with challenging benchmarks. In *Proceedings of the Fifth International Workshop on Designing Meaning Representations @ LREC-COLING 2024*, pages 162–175, Torino, Italia. ELRA and ICCL.

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey.

Jiawei Zhou, Tahira Naseem, Ramón Fernández Astudillo, and Radu Florian. 2021. Amr parsing with action-pointer transformer. In *North American Chapter of the Association for Computational Linguistics*.

A Prompt

The following is a complete example of the prompts we use for the LLMs. Since the models we use are all instruction-based versions, the prompt is structured in a dialogue format.

RASP Prompt

```
user:
Please parse the following text into
Discourse Representation Structure,
considering using the concepts based
on the following glosses:
- birdwatch.v.01:  watch and study
birds in their natural habitat
- saw.n.01:  a condensed but memorable
saying embodying some important fact
of experience that is taken as true
by many people
- saw.n.02:  hand tool having a
toothed blade for cutting
- saw.n.03:  a power tool for cutting
wood
- saw.v.01:  cut with a saw
- harrier.n.01:  a persistent attacker
- harrier.n.02:  a hound that
resembles a foxhound but is smaller;
used to hunt rabbits
- harrier.n.03:  hawks that hunt over
meadows and marshes and prey on small
terrestrial animals
- kite.n.01:  a bank check that has
been fraudulently altered to increase
its face value
- kite.n.02:  a bank check drawn on
insufficient funds at another bank in
order to take advantage of the float
- kite.n.03:  plaything consisting of
a light frame covered with tissue
paper; flown in wind at end of a
string
- kite.n.04:  any of several small
graceful hawks of the family
Accipitridae having long pointed
wings and feeding on insects and
small animals
- kite.v.01:  increase the amount (of
a check) fraudulently
- kite.v.02:  get credit or money by
using a bad check
- kite.v.03:  soar or fly like a kite
- kite.v.04:  fly a kite
- hobby.n.01:  an auxiliary activity
- hobby.n.02:  a child's plaything
consisting of an imitation horse
mounted on rockers; the child
straddles it and pretends to ride
- hobby.n.03:  small Old World falcon
formerly trained and flown at small
birds

Text to parse:  Johanna went
birdwatching.  She saw a harrier, a
kite, and a hobby.

model:
female.n.02 Name "Johanna" time.n.08
TPR now birdwatch.v.01 Agent -2 Time
-1 ELABORATION <1
female.n.02 ANA -3 see.v.01
Experiencer -1 Time +1 Stimulus +3
time.n.08 TPR now
harrier.n.03 kite.n.04 entity.n.01
Sub -2 Sub -1 Sub +1 hobby.n.03
```

Normal Prompt

```
user:
Text to parse:  Johanna went
birdwatching.  She saw a harrier, a
kite, and a hobby.

model:
female.n.02 Name "Johanna" time.n.08
TPR now birdwatch.v.01 Agent -2
Time -1 ELABORATION <1 female.n.02
ANA -3 see.v.01 Experiencer -1 Time
+1 Stimulus +3 time.n.08 TPR now
harrier.n.03 kite.n.04 entity.n.01
Sub -2 Sub -1 Sub +1 hobby.n.03
```

B Experiment Settings

Table 6 and 7 provide the basic details of the experiments and models.

Category	Details	Category	Details
Stage	SFT/inference	Precision	fp16
Fine-tuning	LoRA	Batch Size	1
Cutoff Length	1024	GPU Number	4
Learning Rate	10^{-4}	GPU	H100
Epochs	10	lr scheduler	cosine

Table 6: Configurations for large language models Fine-Tuning and Inference.

Model	Details
BART-large	facebook/bart-large
T5-large	google-t5/t5-large
byT5-large	google/byt5-large
Phi3-4B	microsoft/Phi-3.5-mini-instruct
Qwen2.5-3B	Qwen/Qwen2.5-3B-Instruct
Qwen2.5-7B	Qwen/Qwen2.5-7B-Instruct
LLama3.2-3B	meta-llama/Llama-3.2-3B-Instruct
LLama3.1-8B	meta-llama/Llama-3.1-8B-Instruct
Gemma2-2B	google/gemma-2-2b-it
Gemma2-9B	google/gemma-2-9b-it

Table 7: Details of Models.

C Additional Experiments

We present the results of fine-tuning on Normal data and testing by RASP prompt, as shown in Tables 8 and 9. This approach involves providing retrieval information during inference but using only text-to-DRS data during training. From the results, it is evident that this training method adversely affects the model's performance, particularly on the standard test. We believe that fine-tuning reduces the models' ability of in-context learning, which limits the models from effectively utilizing the additional information provided by retrieval.

Model	Size	Input	Graph-level			Node-level
			Hard-SMatch↑	Soft-SMatch↑	IFR↓	F score↑
Phi3	4B	Normal	85.74	87.92	4.94 (59)	81.60
		RASP	66.78 (–22.1%)	70.88 (–19.4%)	14.9 (178)	63.50 (–22.2%)
Mistral	7B	Normal	89.95	92.48	2.00 (24)	83.90
		RASP	83.22 (–7.5%)	85.90 (–7.1%)	3.58 (43)	80.10 (–4.5%)
Qwen2.5	3B	Normal	86.50	88.64	4.69 (56)	82.60
		RASP	84.32 (–2.5%)	87.44 (–1.4%)	5.00 (60)	81.90 (–0.8%)
	7B	Normal	89.88	91.83	2.51 (30)	84.50
		RASP	86.23 (–4.1%)	90.78 (–1.1%)	2.57 (33)	83.40 (–1.3%)
LLama3	3B	Normal	87.30	90.01	3.34 (40)	81.50
		RASP	85.90 (–1.6%)	86.91 (–3.4%)	4.10 (49)	77.59 (–4.8%)
	8B	Normal	89.92	92.46	2.09 (25)	83.90
		RASP	88.65 (–1.4%)	91.30 (–1.3%)	2.50 (30)	82.11 (–2.1%)
Gemma2	2B	Normal	89.20	91.08	3.01 (36)	84.20
		RASP	84.40 (–5.4%)	89.93 (–1.3%)	3.01 (36)	80.11 (–4.9%)
	9B	Normal	90.72	93.15	1.67 (20)	84.67
		RASP	91.11 (+0.4%)	93.35 (+0.2%)	1.79 (21)	83.10 (–1.9%)

Table 8: Performance on standard test.

Model	Input	Noun	Verb	Modifiers	Overall
phi3-4B	Normal	35.48	36.91	46.97	37.91
	RASP	40.03 (+12.8%)	36.32 (–1.6%)	49.13 (+4.6%)	40.03 (+5.6%)
Mistral-7B	Normal	38.02	40.61	50.00	39.87
	RASP	40.90 (+7.6%)	49.27 (+21.3%)	50.00 (–0.0%)	43.61 (+9.4%)
Qwen2.5-7B	Normal	38.51	37.52	46.97	39.12
	RASP	40.11 (+4.2%)	43.54 (+16.1%)	50.00 (+6.5%)	41.94 (+7.2%)
LLama3.2-8B	Normal	37.06	34.79	47.73	37.59
	RASP	42.10 (+13.6%)	38.88 (+11.8%)	49.00 (+2.7%)	42.00 (+11.7%)
Gemma2-9B	Normal	39.68	45.01	55.30	42.54
	RASP	**45.93** (+15.8%)	**50.11** (+11.3%)	**59.70** (+8.0%)	**48.34** (+13.6%)

Table 9: Performance on the challenge set.

Not Just Who *or* What: Modeling the Interaction of Linguistic and Annotator Variation in Hateful Word Interpretation

Sanne Hoeken[1], **Özge Alaçam**[1], **Dong Nguyen**[2], **Massimo Poesio**[2,3], **Sina Zarrieß**[1]
[1]Bielefeld University, Germany [2]Utrecht University, the Netherlands
[3]Queen Mary University of London, United Kingdom
{sanne.hoeken, oezge.alacam, sina.zarriess}@uni-bielefeld.de
{d.p.nguyen, m.poesio}@uu.nl

Abstract

Interpreting whether a word is hateful in context is inherently subjective. While growing research in NLP recognizes the importance of annotation variation and moves beyond treating it as noise, most work focuses primarily on annotator-related factors, often overlooking the role of linguistic context and its interaction with individual interpretation. In this paper, we investigate the factors driving variation in hateful word meaning interpretation by extending the HateWiC dataset with linguistic and annotator-level features. Our empirical analysis shows that variation in annotations is not solely a function of *who* is interpreting or *what* is being interpreted, but of the interaction between the two. We evaluate how well models replicate the patterns of human variation. We find that incorporating annotator information can improve alignment with human disagreement but still underestimates it. Our findings further demonstrate that capturing interpretation variation requires modeling the interplay between annotators and linguistic content and that neither surface-level agreement nor predictive accuracy alone is sufficient for truly reflecting human variation.[1]

Content warning! Some examples in this paper contain language that may be offensive, for illustrative purposes; we recognize their potential harm.

1 Introduction

Words play a central role in hate speech by encoding derogatory meanings. The meaning of such words is rarely fixed but highly dependent on context and interpretation which poses a significant challenge for both theoretical understanding and computational modeling of hate speech (Sayeed, 2013). Despite growing interest in hate speech detection, there has been little systematic investigation into the semantic and pragmatic mechanisms that underlie how hateful word meanings are interpreted.

Recent work by Hoeken et al. (2024) introduced the HateWiC dataset, which identified substantial variation and disagreement in judgments about whether a word is hateful in context. Models tend to underperform on those cases where annotators disagree. Although incorporating annotator demographic information shows modest improvements in model performance, the underlying sources driving these variations remain poorly understood. This aligns with a broader trend in NLP research, that moves away from aggregated judgments to explicitly modeling inter-annotator variation (Uma et al., 2021; Basile et al., 2021).

Yet, the focus in NLP research on label variation in subjective tasks has largely remained on *who* is interpreting (Kocoń et al., 2021; Orlikowski et al., 2023), with far less attention given to *what* is being interpreted. While linguistic content has always been the basis for classification, recent subjectivity-focused approaches tend to sideline the role of the content itself. Only a few studies acknowledge the role of linguistic ambiguity in subjective labeling (Sandri et al., 2023; Jiang and Marneffe, 2022).

Table 1 illustrates how subjective variation can emerge from both linguistic and annotator features with examples from HateWiC. Variation in perceived hatefulness of the word *napoleon* in the first example likely arises from ambiguity between senses (food vs. person) with limited context. Whereas in the second example annotator differences likely contributed to disagreement, as the annotators seem to have different tendencies to label content as hateful (based on their label ratios). Lastly, the *shrink* example shows that the same annotator's tendency can shift depending on the

[1]Code and supplementary materials for this study are available at https://github.com/SanneHoeken/HateWiCVariation.

Proceedings of the 16th International Conference on Computational Semantics, pages 73–87
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

Word in Context	Term	Sense Definition	Sense Domain	Sense Person Aspect	Context Length	Ann. Id	Gender	Hateful Ratio	Label
Miss Manvers thrust aside a garnished plate and attacked her *napoleon*.	napoleon	Another name for a millefeuille pastry.	Food	NotPerson	11	36 69	Female Female	▲▲▲ ▲▲	✗ ✓
He is the *napoleon* of crime, Watson. He is the organizer of half that is evil [...]	napoleon	A person having come to dominate an area of activity through illegality.	Person	Personality/ behavior	41	36 75	Female Male	▲▲▲ ▲	✗ ✓
My *shrink* said that he was an enabler, bad for me.	shrink	A psychiatrist or psychotherapist.	Person	Profession	11	36 4	Female Female	▲▲▲ ▲▲▲	✓ ✗

Table 1: Examples from the HateWiC dataset, with augmented linguistic and annotator information, that illustrate how label variation (✗= *hateful*; ✓= *not hateful*) can arise from linguistic ambiguity (e.g. different senses of napoleon) as well as from annotator tendencies (Hatefulness Ratio from *low* (▲) to *high* (▲▲▲)), while also highlighting the interaction of these features with subjective interpretation.

linguistic content they are judging, such as whether the term's referent is defined by profession or behavior (Person Aspect). It is this interaction between linguistic features and subjective tendencies that shapes variation in interpretation.

Within the ongoing search for meaningful predictors of human variation in subjective language interpretation, relatively little attention has been given to the level of word meaning. Moreover, most studies only focus on annotator-related features, neglecting the interplay between semantics, linguistic context, and subjective interpretation that shapes how hateful meanings arise. Additionally, existing modeling efforts typically emphasize overall performance metrics without assessing whether models replicate the *patterns* of human variation. Yet understanding and modeling such patterns is crucial for NLP systems to meaningfully reflect the subjective nature of language interpretation in sensitive tasks like hate speech detection.

Addressing these gaps, we augment the HateWiC dataset with linguistic and annotator-level features (§3) and empirically show that variation in hateful meaning interpretation is driven not just by who the annotator is or what is being annotated, but by their interaction (§4). Building on this analysis, we propose an evaluation framework that assesses whether BERT-based classification models capture this variation (§5). The results (§6) demonstrate that while models incorporating annotator-specific inputs can reproduce superficial variation, they substantially underestimate its magnitude and fail to capture the internal structure of variation found in human annotations.

2 Related Work

In what follows, we discuss prior work on hateful word meaning in NLP and subjective variation in Hate Speech Detection (HSD), both of which motivate our study.

2.1 Hateful word meaning in NLP

Capturing variation in word meaning has long been a focus in NLP (Pustejovsky, 1991; Schütze, 1998; Haber and Poesio, 2024). Computational approaches to lexical semantics have included tasks such as Word Sense Disambiguation (Loureiro et al., 2021), Word Sense Induction (Eyal et al., 2022) and Lexical Semantic Change Detection (Periti and Montanelli, 2024). Methods predominantly rely on embedding-based techniques using encoder-based language models and often employ contextualized sense similarity metrics (Blevins and Zettlemoyer, 2020; Cassotti et al., 2023). Moreover, the tasks and approaches typically depend on general-purpose resources and corpora that are oriented toward standard language usage. Consequently, they tend to focus on denotative rather than domain-specific or connotative meaning (Potts, 2007) (e.g. capturing denotative shifts as with a word like *plane* changing from primarily a geometric concept to also denoting an aircraft, in contrast to connotative shifts, such as *spinster* becoming more negatively charged over time).

In contrast, some work has addressed connotative meaning in the context of hate speech by examining lexical features used in sequence-level detection (Koufakou et al., 2020; Zampieri et al., 2022). Other studies have explored the disambiguation and detection of such terms, including subtle forms like dog whistles (Kruk et al., 2024; Mendelsohn et al., 2023). Prior research has also examined more clear-cut cases, such as swear words (Pamungkas et al., 2022) and slurs (Hoeken et al., 2023), which are often argued to be more stable across contexts (Frigerio and Tenchini, 2019). Additional work has addressed more ambiguous pejorative terms (Dinu et al., 2021). However, much of this research adopts a (binary) classification perspective, with limited attention to intra-word variation, i.e. how the connotative meaning of a term shifts across con-

texts or individuals. Recently, Hoeken et al. (2024) addressed this issue with the introduction of the HateWiC dataset. Their findings highlight the substantial variation in how hateful word meanings are perceived, but the question about what underlies this variation remains.

2.2 Subjective variation in HSD

Annotator disagreement is increasingly recognized as a signal of subjective variation rather than mere labeling noise (Larimore et al., 2021; Plank, 2022; Fleisig et al., 2024). This shift is especially pertinent in HSD, where personal differences strongly influence interpretive judgments. Several studies have highlighted the role of annotator identity in shaping perceived offensiveness. While some highlight the relevance of sociodemographic variables like gender and age (Kocoń et al., 2021; Sang and Stanton, 2022), recent findings suggest that such variables often act as noisy proxies and are poor predictors for interpretation variation (Alipour et al., 2024; Orlikowski et al., 2023). Several studies consider other annotator factors like ideology (Sap et al., 2022) or moral values (Mostafazadeh Davani et al., 2024), yet all consider annotator information as the primary source of variation.

Recent modeling approaches have incorporated annotator-specific information in various ways. These include demographic-based embeddings (Fleisig et al., 2023), embeddings based on annotator ids or label histories (Deng et al., 2023; Mokhberian et al., 2024), and label distribution learning (Weerasooriya et al., 2023). Other recent personalization techniques involve multimodal signals like gaze (Alacam et al., 2024), or fine-tuning LLMs with annotator-specific prompts (Orlikowski et al., 2025). Despite these advances, most efforts emphasize improvements in predictive performance, often evaluated via accuracy metrics. An exception is Anand et al. (2024), who propose aligning model confidence with annotator agreement as a step toward more human-aligned predictions.

Our work contributes to this line of research by explicitly modeling individual variation in hateful word interpretation, and evaluating models by how well they capture the *structure* of this variation across linguistic and annotator-related dimensions.

3 Data & Features

To analyze variation in the interpretation of potentially hateful words, we use the HateWiC dataset (Hoeken et al., 2024), which provides contextualized word usages annotated for perceived hatefulness. We further enrich this dataset with additional linguistic and annotator-related features to facilitate a comprehensive empirical analysis of variation.

3.1 The HateWiC dataset

The HateWiC dataset comprises approx. 4,000 word-in-context (WiC) instances, each annotated independently by three annotators (N $\approx$ 12k total annotations). Annotation was distributed across 48 annotators, with each annotating 250 instances. Each instance consists of a target term embedded in a sentence and linked to a Wiktionary definition that corresponds to its contextual meaning (totaling 1,888 unique definitions). This setup thus provides sense-level information. The terms included have at least one sense referring to people and considered offensive based on Wiktionary data. Annotators were asked to indicate whether the meaning of the target term in the specific sentence was hateful or not, and could also indicate undecided. The dataset is balanced across the two main classes.

To measure variation, we use a binary variable indicating whether an individual annotator's label matches the majority label for that instance (agree) or not (disagree). We adopt this annotation-level operationalization because it allows us to associate both linguistic features of the text and annotator features (which require individual annotations) with variation in interpretation. We further augment the HateWiC data with various supplementary features, described (and highlighted in bold) below.

3.2 Linguistic features

We manually annotated the semantic **Domain** of each Wiktionary definition, assigning categories such as Person, Animal, and Food. This is motivated by the idea that ambiguity across these broad semantic domains (e.g. Napoleon as a person versus a dessert) may lead to variation in hateful interpretation. We further annotated the **Person Aspect** emphasized, distinguishing among categories such as Personality/Behavior, Ethnicity/Nationality and Appearance. These dimensions could influence annotators' judgments of hatefulness differently. For example, references to ethnicity may evoke stronger perceptions of offense compared to those focused on behavior or appearance. All annotations were carried out by two linguistic experts, with full dual annotation for validation. More details on category taxonomies and annotation are provided in

Appendix A.

In addition to these semantic annotations, we included the **part of speech (POS)** linked to each sense definition, which was already included as metadata in the HateWiC dataset. We also consider for each word in context the **Context Length**, measured by the number of whitespace-separated tokens, as shorter contexts might provide fewer clues for disambiguation which potentially increases disagreement among annotators.

Finally, we incorporate the **Grammatical Role** of the target word in its context. Grammatical Role was identified using SpaCy's dependency parser and mapped to a coarser set of ten categories such as subject, object and preposition (fully specified in Appendix A). This syntactic information might affect how strongly a term is emphasized and thereby influence variation in perceived hateful intent.

3.3 Annotator features

We incorporated annotator-related features by leveraging information already present in the HateWiC dataset. This includes the **Annotator Id**, along with available sociodemographics (**Gender**, **Ethnicity**, and **Age**). We converted absolute age values into age categories (e.g. '20-29'). As an additional feature, we computed each annotator's **Hatefulness Ratio**, defined as the proportion of instances they labeled as hateful across the dataset (see also Appendix A). This metric serves as an approximation of an annotator's tendency to classify content as hateful.

4 Empirical Analysis

We begin our empirical analysis by assessing the overall degree of annotator agreement in the HateWiC dataset. We calculate inter-annotator agreement on the original dataset's annotations, with Krippendorff's alpha resulting in 0.452. This value reflects moderate agreement and matches the original HateWiC paper's findings (Hoeken et al., 2024)[2]. Moving beyond surface-level agreement, we statistically test the association between our enriched set of linguistic and annotator features, and the binary outcome of agreement with the majority.

4.1 Independent feature associations

For a fair comparison of statistical test outputs, we converted numerical features (Context Length and Hatefulness Ratio) into categorical variables using quantile-based binning (with n_bins = 4). We conducted Chi-squared tests of independence to assess the relationship between each feature and annotation agreement (i.e. agree or disagree with the majority vote). Effect sizes were calculated using Cramer's V to measure the strength of associations.

Type	Feature	χ^2	p-value	Cramer's V
linguistic	Person Aspect	61.43	<0.001	0.072
	Domain	31.83	<0.001	0.052
	Context Length	48.53	<0.001	0.064
	Grammatical Role	18.99	0.040	0.040
	POS	4.06	0.669	0.018
annotator	Annotator Id	238.11	<0.001	0.141
	Hatefulness Ratio	37.32	<0.001	0.056
	Ethnicity	59.39	0.000	0.071
	Age	14.53	0.006	0.035
	Gender	4.73	0.094	0.020

Table 2: Statistical test results for association of categorical features with annotation agreement (agree or disagree with majority vote)

The results in Table 2 show several statistically significant associations. Among linguistic features, Person Aspect shows the strongest association. Context Length and Domain also have significant effects on the agreement ($p < 0.001$) and Grammatical Role is marginally significant ($p = 0.04$). In the annotator-related features, Annotator Id shows the strongest association. Ethnicity and Hatefulness Ratio are also significant ($p < 0.001$). Age is significant at the 0.01 level. Further details on the computation and results, including contingency tables, are provided in Appendices B and D.

Overall, the analysis indicates that both linguistic properties of the input and demographic/behavioral characteristics of annotators influence annotation variation, with the strongest effects observed at the annotator level. While many features have significant effects, the effect sizes are generally small (Cramer's V < 0.15), indicating weak to modest associations. This suggests that a large portion of variation in annotation variation remains unexplained by these main effects.

4.2 Feature interaction associations

Figure 1 displays both individual and pairwise interaction effects on annotation agreement, again based on Chi-squared tests, this time considering combinations of two features as well. The diagonal represents individual feature effects, while the off-diagonal quadrants correspond to pairwise interactions: the lower-left quadrant shows interactions between linguistic and annotator features, the

[2]The alpha value reported in Hoeken et al. (2024) was obtained without considering the undecided label, a difference that does not appear to substantially affect the outcome.

upper-left linguistic × linguistic interactions, and the lower-right annotator × annotator interactions.

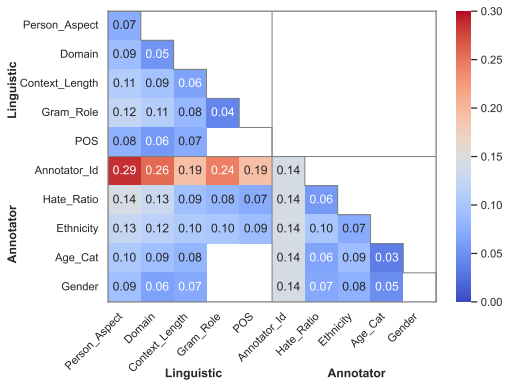


Figure 1: Heatmap of Cramer's V effect sizes showing both individual and pairwise associations of features with annotation agreement. The upper triangle (above the diagonal) as well as non-significant ($p > 0.05$) interaction effects are masked.

Generally, interactions explain more variation in annotation agreement than individual features. Particularly, interactions between annotator and linguistic features are the strongest, with the highest effect size of V = 0.29 for Person Aspect × Ann Id. This pattern of low main effect but high cross-type (linguistic × annotator) interaction supports that annotation variation is more a function of who is interpreting what, rather than just who, or what.

Within type interactions, the higher interaction effects among linguistic features (max V = 0.12 for Person Aspect × Grammatical Role), compared to individual features (max V = 0.07), emphasize that the combined effect of linguistic features matters more for meaning variation, which aligns with linguistic theories of compositionality and context-dependent meaning (Partee et al., 1984).

Adding interactions among annotator features does not increase association strength beyond what is captured by Annotator Id alone. This is logical because Annotator Id essentially encapsulates all annotator-related factors. Ignoring Annotator Id, interactions among other annotator features show modestly stronger effects than individual features, with the ethnicity × Hatefulness Ratio interaction yielding V = 0.10. This implies possible interpretative biases (reflected by tendency to label hate) linked to cultural context. Nonetheless, these effects remain smaller than those involving Annotator Id, thus the results show that individual annotator differences beyond demographics and labeling tendency has stronger influence on the agreement.

4.3 A closer look: Person Aspect × Hatefulness Ratio

While statistical tests and interaction analyses provide evidence of feature associations with annotation agreement, inspecting the directions and patterns of these effects allows for a more concrete interpretation. We illustrate this by zooming in on the interaction effect of two features from our analysis. Figure 2 visualizes the interaction between the semantic Person Aspect of the target word and annotators' hateful labeling tendency (Person Aspect × Hatefulness Ratio, with the latter discretized into four intervals. The disagreement probability shows distinct patterns across Person Aspect categories. For example, instances in the Appearance or Social class categories exhibit relatively high disagreement for annotators with a low Hatefulness Ratio and less disagreement with moderate to high ratios. Conversely, the Kinship/social category exhibits the opposite trend. These diverging patterns emphasize that annotator tendencies do not exert uniform effects across linguistic categories. Instead, the influence of individual biases on annotation variation is mediated by the specific semantic characteristics of the content.

5 Computational Modeling

In this section, we investigate to what extent computational models with different inputs can capture human variation in annotations. We address this question in the context of the binary classification task that predicts the individual annotations in the HateWiC dataset (12K annotations of words in context, labeled hateful or not hateful based on their meaning in that context). We explicitly model and analyze this variation by conditioning predictions on auxiliary inputs such as annotator identity or demographics. The primary goal is to gain insights into alignment with human interpretation variation rather than optimize benchmark performance.

5.1 Model architecture & experiments

We largely follow the approach proposed by Deng et al. (2023), who incorporate annotator embeddings into a BERT model. Their mechanism relies on a predefined annotator id vocabulary. We extend this approach by introducing a modular framework that allows integration of *auxiliary* information, including not only discrete id-based inputs but also free-form text descriptions, alongside standard input text (*primary* input). The model architecture

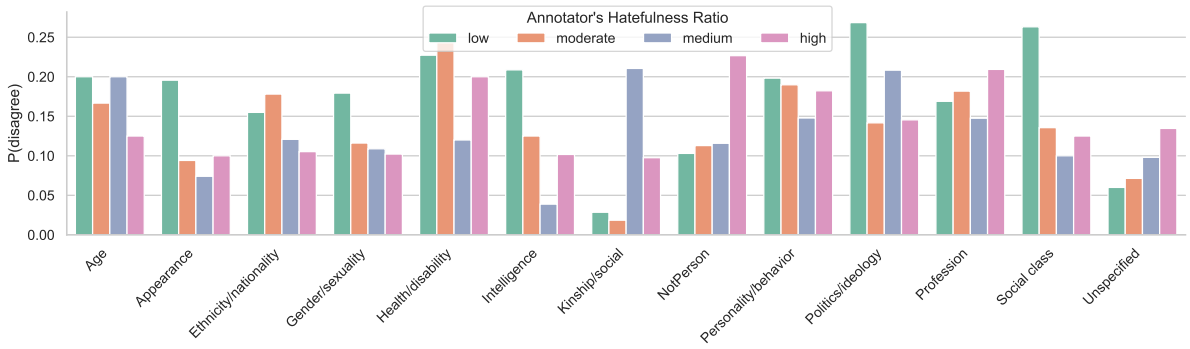


Figure 2: Interaction between the person-related semantic category of the target word (Person Aspect) and annotator's individual tendency to label instances as hateful (Hatefulness Ratio) on the probability of disagreement as the proportion of annotations where individual annotators disagreed with the majority vote.

builds upon a pre-trained encoder for representations of textual inputs. Specifically, we initialize all models with the base version of ModernBERT (Warner et al., 2024) as encoder. Similar to Deng et al. (2023) we adopt a learnable feature-wise weighting mechanism for auxiliary embeddings.

Primary text embeddings For each HateWiC instance, the primary input is the sentence containing the target term (**WiC**). Alternatively, following Hoeken et al. (2024), we test replacing this input with the corresponding Wiktionary definition (**WikDef**), or using a concatenation of both. WikDef provides lexical semantic information about the term in a non-contextualized form. Each input type is independently passed through the encoder to obtain a [CLS] representation, which serves as the primary feature embedding.

Auxiliary annotator embeddings Following Deng et al. (2023), an embedding layer maps auxiliary ids to dense vectors, which are jointly trained with the rest of the model, yielding *id-based* annotator embeddings (**ann id**). We extend this framework by enabling auxiliary inputs in natural language form, resulting in *text-based* annotator embeddings. These include: (i) annotator ids (**ann id**) expressed as text (e.g. "annotator_12"), (ii) a description of demographic characteristics (**profile descr.**) (e.g. "The reader is Female, Asian and 28") and (iii) a description of a single characteristic, for which we specifically test **ethnicity** (e.g. "Asian"). Additionally, inspired by recommender system approaches (Shin et al., 2023), we explore (iv) representations of each annotator's label history (**ann. behavior**) as the set of prior WiC instances they labeled as hateful (drawn from the training set). All textual inputs are processed using the same ModernBERT encoder, with the [CLS] token representation used as the embedding. For behavior-based inputs, which consist of a list of texts, the [CLS] representations are averaged to produce a single embedding.

Feature Weighted Classifier (FWC) To integrate the auxiliary embeddings with the primary text representation, we adopt a feature-wise learnable weighting scheme. Each auxiliary embedding is assigned a scalar weight (learned during training) that determines its contribution. The weighted auxiliary vectors are then concatenated with the primary text embedding and passed into the classifier. The classifier is a single-layer multilayer perceptron (MLP) comprising a linear transformation, ReLU activation, dropout regularization, and a final linear layer mapping to the output classes.

Experimental setup We evaluate ten model configurations: three using only primary inputs (i.e. the WiC and/or its definition), and seven that additionally incorporate auxiliary annotator information. Model predictions are generated for each individual annotation in the HateWiC dataset using a 10-fold cross-validation framework. Each fold follows a fixed 80-10-10 split into training, validation, and testing sets. Further implementation details, including libraries, hyperparameters, and hardware specifics, are provided in Appendix C.

5.2 Evaluation

Our goal is to assess how closely computational models capture human variation in annotation for the HateWiC task. In the previous section, we statistically analyzed a range of linguistic and annotator-specific features to understand their influence on human agreement. Here, we evaluate

whether models can replicate these patterns by analyzing their predictions of individual annotations (typically three per sentence), with and without annotator-specific information as auxiliary input. In the latter case, models simulate predictions from annotators by conditioning on annotator identity.

Prediction agreement To quantify how closely the model's predictions resemble human annotation variation in terms of inter-annotator agreement measured through Krippendorff's alpha (α), we define an **Agreement Alignment** score as:

$$\mathbf{AA} = 1 - |\alpha_{\text{model}} - \alpha_{\text{human}}|$$

Here, α_{model} is computed over the model's predicted annotations and thus reflects the model's variation across simulated annotators. α_{human} is the alpha from actual human annotations. The score ranges from 0 to 1, with higher values indicating that the degree of variation in the model's predictions more closely matches the degree of variation observed in human annotations.

Agreement patterns To assess whether models go beyond surface-level agreement and replicate deeper variation patterns, we examine whether they reproduce the same effects of linguistic and annotator variables on label variation as observed in human data. Specifically, we conduct the same statistical tests (§4), replacing human annotations with model predictions. Variation is again treated as a binary variable (agree or disagree) based on whether each individual model prediction aligns with the model-level majority vote. This mirrors the human data procedure, where individual annotations were compared to the human majority.

Using the same set of ten linguistic and annotator features listed in Table 2, we examine both main effects of individual features (10 effects) and interactions between feature pairs, i.e. 45 effects from all possible pairwise combinations. To quantify how closely a model replicates variation patterns, we compute the **Relative Pattern Alignment (RPA)** score between human and model effect sizes (measured using Cramér's V) across all n effects, which we define as:

$$\mathbf{RPA} = 1 - \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\text{effect}_{\text{human},i} - \text{effect}_{\text{model},i}}{\text{effect}_{\text{human},i}}\right|$$

We normalize each error by the magnitude of the corresponding human effect size to accommodate the small magnitude of Cramér's V and to prevent larger effects from disproportionately influencing the score. The final metric is inverted so that higher RPA values indicate stronger alignment between the model's and humans' variation patterns.

Prediction accuracy Finally, we directly compare the model predictions to individual human annotations, following traditional evaluation practices. For each model, we report **accuracy** across all instances.

6 Results & Discussion

We present results for all ten model configurations in Table 3, which vary in terms of their input features: (i) primary input only, (ii) primary input with *id-based* annotator embeddings, and (iii) primary input with *text-based* annotator embeddings.

FWC config.	model input	AA	RPA	Acc.
primary only	WiC	0.452	0.000	0.650
	WikDef	0.452	0.000	0.671
	WiC + WikDef	0.452	0.000	0.700
+ aux. (*id-based*)	WiC + ann. id	0.670	0.620	0.664
	WikDef + ann id	**0.732**	0.632	0.682
	WiC + WikDef + ann. id	0.638	**0.658**	**0.704**
+ aux. (*text-based*)	WiC + ann. id	0.516	0.567	0.656
	WiC + profile descr.	0.576	0.557	0.654
	WiC + ethnicity	0.501	0.539	0.654
	WiC + ann. behavior	0.452	0.000	0.654

Table 3: Agreement Alignment score, Relative Pattern Alignment score and accuracy for the different model configurations compared to the human annotation data.

6.1 Prediction agreement

Models that process only primary text naturally produce identical predictions across simulated annotators for each instance. This results in perfect inter-annotator agreement ($\alpha_{\text{model}} = 1$). Consequently, they score lowest on Agreement Alignment (AA = 0.452), as they fail to reproduce the human variation in annotations. In contrast, models that incorporate auxiliary annotator information, particularly those with *id-based* embeddings, exhibit lower agreement rates. This indicates that simulated annotators produce diverging predictions on the same primary input, mimicking the variation observed in human annotations.

Text-based auxiliary inputs result only in modest improvements over primary-only baselines and underperform compared to *id-based* embeddings. For instance, using *text-based* annotator ids yields an AA of 0.516, whereas the corresponding *id-based* configuration achieves 0.670. These differences

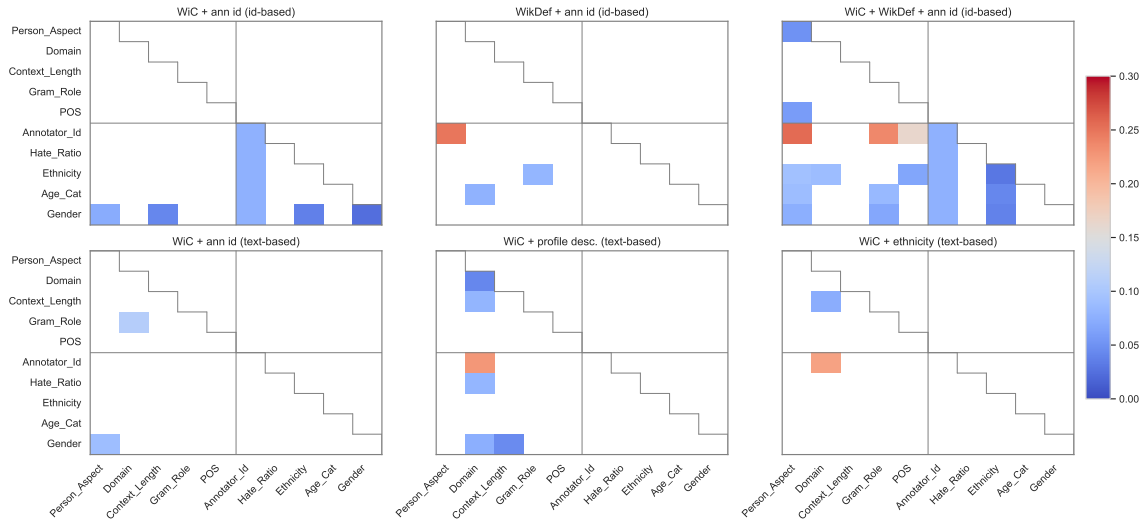


Figure 3: Heatmaps of Cramer's V effect sizes showing both individual (along the diagonal) and pairwise associations of features with model prediction agreement for different FWC model configurations (named after their inputs). The upper triangles (above the diagonal) as well as non-significant ($p > 0.05$) interaction effects are masked.

might originate from the fact that *id-based* embeddings are jointly trained, letting the model distinguish the annotators in a more clear-cut manner, whereas *text-based* inputs rely on static representations from a pre-trained encoder, limiting their influence on the model's decision making. Notably, the WiC + ann. behavior model maintains perfect inter-annotator agreement ($\alpha_{\text{model}} = 1$), suggesting that the behavior representations do not inject any variation into model predictions. A possible explanation is that each annotator's behavior embedding is a fixed average of the hateful sentences they labeled, which may smooth out fine-grained differences and lack strong signals to distinguish annotators.

Overall, these findings suggest that conditioning on annotator identity introduces label variation, but the way this auxiliary input is provided affects the extent of this variation. Yet, in general, models underestimate the magnitude of variation observed in human annotations.

6.2 Agreement patterns

While Agreement Alignment quantifies whether models produce human-like variation in an aggregated manner, it does not capture *how* that variation arises. To probe this, we analyze Relative Pattern Alignment (RPA), which measures how well a model replicates the internal structure of human variation. High AA does not always translate to high RPA, indicating that the variation in human data and model predictions might originate from different instances. For example, while the model with WikDef + ann. id has the highest AA (0.732), the configuration with combined inputs (WiC + WikDef + ann. id) achieves the best RPA (0.658). These results reveal that surface-level agreement can be misleading, since it does not guarantee alignment with the internal structure of human variation.

Figure 3 visualizes feature association patterns for each of the six models, restricted to those exhibiting variation in their predictions ($\alpha_{\text{model}} < 1$). It displays for each model a heatmap of Cramer's V effect sizes showing both individual and pairwise associations of features with prediction agreement. The human annotation data showed a diverse range of significant effects (48 out of 55 tested), including interactions between annotator features, linguistic features, and cross-type combinations. Among these, the latter were particularly prominent. Models generally captured far fewer significant effects and vary widely in their replication of human-like effect structures. A key distinction emerges in the types of feature interactions that models are able to replicate. The best model in terms of RPA (WiC + WikDef + ann. id) captures numerous significant effects spanning all three interaction types. In contrast, only two significant effects were identified for the model with WiC + ann id (*text-based*) inputs, none involving annotator × annotator interactions.

Overall, these findings show the importance of not just measuring agreement rates, but also systematically analyzing the patterns of variation, which can offer a more fine-grained view of how model predictions reflect the structure of human annota-

tion behavior.

6.3 Prediction performance

Across all configurations, predictive accuracy remains relatively stable (0.65–0.70). The highest accuracies are observed for models using semantic-rich inputs, i.e. including both sentence context (WiC) *and* definitions (WikDef) as inputs. This highlights the importance of linguistic information for predicting individual annotations and aligns with our earlier findings on the role of linguistic features in human annotations. In addition, models that best capture human-like variation do not necessarily predict individual labels more accurately. For instance, although the WikDef + ann. id model exhibits strong AA (0.732) and RPA (0.632), its accuracy (0.682) is only marginally better than primary-only models. These findings suggest that optimizing for predictive accuracy and optimizing for alignment with human variation may constitute distinct modeling objectives that warrant separate consideration in model development.

7 Conclusion

In this paper we demonstrated that the variation in interpretation of hateful word meaning is not merely a function of *who* the annotator is or *what* is being annotated, but of the interaction between the two. Through empirical analysis of the HateWiC dataset, we showed that both linguistic properties of the target word in context and annotator characteristics shape interpretive variation. Our evaluation of model alignment with human variation further reveals that although models that incorporate annotator-specific information introduce human-like variation at a surface level, they still underestimate the magnitude of variation observed in human annotations and generally fail to represent the internal structure of variation. In conclusion, our findings show that capturing human interpretive variation requires modeling the interaction between annotators and linguistic content, and that surface-level agreement or predictive accuracy alone does not ensure true alignment with human variation.

Limitations

Alongside its contributions, this study has several limitations that should be acknowledged:

Binary operationalization. Our analysis relies on binary categorizations for hatefulness (hateful vs. not hateful) and annotator agreement (agree vs. disagree with majority). While this simplifies modeling and interpretation, it risks oversimplifying the complexity of human judgments. Future work could explore multi-class or continuous scales to capture finer distinctions in hatefulness and annotation variation.

Feature selection & categorization. The linguistic and annotator features included in our study, although carefully chosen to cover key linguistic and annotator dimensions, represent a subset of potentially relevant factors. Additionally, some features were either provided in broad categories or grouped during analysis to facilitate reliable statistical analysis. Other linguistic phenomena, richer annotator identity information and more refined categorizations might further explain variation patterns.

Label variation as interpretation variation. We interpret label variation among annotators as indicative of variation in meaning interpretation. While this is a reasonable assumption, other sources of disagreement, such as sloppy annotations or uncertainty, cannot be fully ruled out (e.g. Sandri et al. (2023)). Incorporating complementary data such as annotator confidence ratings or qualitative feedback could strengthen this.

Automatic parsing. The Grammatical Role feature was derived using automatic dependency parsing (SpaCy) without additional validation tailored to the specific dataset. While SpaCy generally offers robust performance, parsing errors could introduce noise in the linguistic feature set. Dataset-specific parser evaluation could improve feature reliability in future analyses.

Data size and imbalance. Some feature categories have limited observations, restricting the use of complex models like mixed-effects regression with random intercepts for annotators. These models treat each subcategory as a separate binary feature which requires enough data per subcategory to produce reliable estimates of variation and interaction effects. Due to this, we relied on Chi-squared tests and effect size measures better suited to the dataset. Larger, more balanced data would enable exploring richer feature effects.

Limited modeling diversity. The modeling component of this study focused on one type of architecture (ModernBERT-based encoder models with auxiliary feature integration). While this design allowed us to systematically evaluate the contribution

of annotator information within a controlled setup, it does not explore the full range of potentially useful architectures. Future work could broaden this scope to assess generalizability across modeling paradigms.

Ethics Statement

Our work builds upon the HateWiC dataset by enriching it with additional linguistic annotations and computational analyses. Apart from the supplementary linguistic annotations (see also Appendix A), no new human annotations were collected for this research beyond what is already available in HateWiC, and no personally identifying information was processed or used. Where annotator identity is used for modeling purposes, it is limited to anonymous identifiers that cannot be traced to real individuals. We recognize that demographic categories such as ethnicity, gender and age provide only a limited representation of individual identity. These features are used here solely to explore variation in annotator interpretation and not to make generalizations about any group.

Given the sensitive nature of hate-related content, we have taken care to conduct our analyses and reporting in a manner that avoids harm. The focus of our work is on variation in interpretation rather than the endorsement or rejection of any specific viewpoint. Our goal is to improve understanding of the variation inherent to such subjective annotation tasks, in order to support the development of computational methods that better account for subjective variation and promote fairness in NLP applications.

Acknowledgments

The authors acknowledge financial support by the project "SAIL: SustAInable Life-cycle of Intelligent Socio-Technical Systems" (Grant ID NW21-059A), which is funded by the program "Netzwerke 2021" of the Ministry of Culture and Science of the State of North Rhine-Westphalia (Germany) and by the project "Dealing with Meaning Variation in NLP", which is funded by the Dutch Research Council (NWO) through the AiNed Fellowship Grant (NGF.1607.22.002).

References

Özge Alacam, Sanne Hoeken, and Sina Zarrieß. 2024. Eyes don't lie: Subjective hate annotation and detection with gaze. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 187–205, Miami, Florida, USA. Association for Computational Linguistics.

Shayan Alipour, Indira Sen, Mattia Samory, and Tanushree Mitra. 2024. Robustness and confounders in the demographic alignment of llms with human perceptions of offensiveness.

Abhishek Anand, Negar Mokhberian, Prathyusha Kumar, Anweasha Saha, Zihao He, Ashwin Rao, Fred Morstatter, and Kristina Lerman. 2024. Don't blame the data, blame the model: Understanding noise and bias when learning from subjective annotations. In *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024)*, pages 102–113, St Julians, Malta. Association for Computational Linguistics.

Valerio Basile, Michael Fell, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, Massimo Poesio, and Alexandra Uma. 2021. We need to consider disagreement in evaluation. In *Proceedings of the 1st Workshop on Benchmarking: Past, Present and Future*, pages 15–21, Online. Association for Computational Linguistics.

Terra Blevins and Luke Zettlemoyer. 2020. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1006–1017, Online. Association for Computational Linguistics.

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.

Naihao Deng, Xinliang Zhang, Siyang Liu, Winston Wu, Lu Wang, and Rada Mihalcea. 2023. You are what you annotate: Towards better models through annotator representations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12475–12498, Singapore. Association for Computational Linguistics.

Liviu P. Dinu, Ioan-Bogdan Iordache, Ana Sabina Uban, and Marcos Zampieri. 2021. A computational exploration of pejorative language in social media. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3493–3498, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Matan Eyal, Shoval Sadde, Hillel Taub-Tabib, and Yoav Goldberg. 2022. Large scale substitution-based word sense induction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4738–4752,

Dublin, Ireland. Association for Computational Linguistics.

Eve Fleisig, Rediet Abebe, and Dan Klein. 2023. When the majority is wrong: Modeling annotator disagreement for subjective tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6715–6726, Singapore. Association for Computational Linguistics.

Eve Fleisig, Su Lin Blodgett, Dan Klein, and Zeerak Talat. 2024. The perspectivist paradigm shift: Assumptions and challenges of capturing human labels. 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 2279–2292, Mexico City, Mexico. Association for Computational Linguistics.

Aldo Frigerio and Maria Paola Tenchini. 2019. Pejoratives: a classification of the connoted terms. *Rivista Italiana di Filosofia del Linguaggio*, 13(1).

Janosch Haber and Massimo Poesio. 2024. Polysemy—evidence from linguistics, behavioral science, and contextualized language models. *Computational Linguistics*, 50(1):351–417.

Sanne Hoeken, Sina Zarrieß, and Ozge Alacam. 2023. Identifying slurs and lexical hate speech via lightweight dimension projection in embedding space. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 278–289, Toronto, Canada. Association for Computational Linguistics.

Sanne Hoeken, Sina Zarrieß, and Özge Alacam. 2024. Hateful word in context classification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 172–186, Miami, Florida, USA. Association for Computational Linguistics.

Nan-Jiang Jiang and Marie-Catherine de Marneffe. 2022. Investigating reasons for disagreement in natural language inference. *Transactions of the Association for Computational Linguistics*, 10:1357–1374.

Jan Kocoń, Alicja Figas, Marcin Gruza, Daria Puchalska, Tomasz Kajdanowicz, and Przemysław Kazienko. 2021. Offensive, aggressive, and hate speech analysis: From data-centric to human-centered approach. *Information Processing Management*, 58(5):102643.

Anna Koufakou, Endang Wahyu Pamungkas, Valerio Basile, and Viviana Patti. 2020. HurtBERT: Incorporating lexical features with BERT for the detection of abusive language. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 34–43, Online. Association for Computational Linguistics.

Julia Kruk, Michela Marchini, Rijul Magu, Caleb Ziems, David Muchlinski, and Diyi Yang. 2024. Silent signals, loud impact: LLMs for word-sense disambiguation of coded dog whistles. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12493–12509, Bangkok, Thailand. Association for Computational Linguistics.

Savannah Larimore, Ian Kennedy, Breon Haskett, and Alina Arseniev-Koehler. 2021. Reconsidering annotator disagreement about racist language: Noise or signal? In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, pages 81–90, Online. Association for Computational Linguistics.

Daniel Loureiro, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2021. Analysis and evaluation of language models for word sense disambiguation. *Computational Linguistics*, 47(2):387–443.

Julia Mendelsohn, Ronan Le Bras, Yejin Choi, and Maarten Sap. 2023. From dogwhistles to bullhorns: Unveiling coded rhetoric with language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15162–15180, Toronto, Canada. Association for Computational Linguistics.

Negar Mokhberian, Myrl Marmarelis, Frederic Hopp, Valerio Basile, Fred Morstatter, and Kristina Lerman. 2024. Capturing perspectives of crowdsourced annotators in subjective learning tasks. 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 7337–7349, Mexico City, Mexico. Association for Computational Linguistics.

Aida Mostafazadeh Davani, Mark Diaz, Dylan K Baker, and Vinodkumar Prabhakaran. 2024. D3CODE: Disentangling disagreements in data across cultures on offensiveness detection and evaluation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18511–18526, Miami, Florida, USA. Association for Computational Linguistics.

Matthias Orlikowski, Jiaxin Pei, Paul Röttger, Philipp Cimiano, David Jurgens, and Dirk Hovy. 2025. Beyond demographics: Fine-tuning large language models to predict individuals' subjective text perceptions.

Matthias Orlikowski, Paul Röttger, Philipp Cimiano, and Dirk Hovy. 2023. The ecological fallacy in annotation: Modeling human label variation goes beyond sociodemographics. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1017–1029, Toronto, Canada. Association for Computational Linguistics.

Endang Wahyu Pamungkas, Valerio Basile, and Viviana Patti. 2022. Investigating the role of swear words in abusive language detection tasks. *Language Resources and Evaluation*, 57(1):155–188.

Barbara Partee et al. 1984. Compositionality. *Varieties of formal semantics*, 3:281–311.

Francesco Periti and Stefano Montanelli. 2024. Lexical semantic change through large language models: a survey. *ACM Comput. Surv.*, 56(11).

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.

Christopher Potts. 2007. The expressive dimension. *Theoretical Linguistics*, 33(2):165–198.

James Pustejovsky. 1991. The Generative Lexicon. *Computational Linguistics*, 17(4):409–441.

Marta Sandri, Elisa Leonardelli, Sara Tonelli, and Elisabetta Jezek. 2023. Why don't you do it right? analysing annotators' disagreement in subjective tasks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2428–2441, Dubrovnik, Croatia. Association for Computational Linguistics.

Yisi Sang and Jeffrey Stanton. 2022. The origin and value of disagreement among data labelers: A case study of individual differences in hate speech annotation. In *Information for a Better World: Shaping the Global Future: 17th International Conference, IConference 2022, Virtual Event, February 28 – March 4, 2022, Proceedings, Part I*, page 425–444, Berlin, Heidelberg. Springer-Verlag.

Maarten Sap, Swabha Swayamdipta, Laura Vianna, Xuhui Zhou, Yejin Choi, and Noah A. Smith. 2022. Annotators with attitudes: How annotator beliefs and identities bias toxic language detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5884–5906, Seattle, United States. Association for Computational Linguistics.

Asad Sayeed. 2013. An opinion about opinions about opinions: subjectivity and the aggregate reader. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 691–696, Atlanta, Georgia. Association for Computational Linguistics.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Kyuyong Shin, Hanock Kwak, Wonjae Kim, Jisu Jeong, Seungjae Jung, Kyungmin Kim, Jung-Woo Ha, and Sang-Woo Lee. 2023. Pivotal role of language modeling in recommender systems: Enriching task-specific and task-agnostic representation learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1146–1161, Toronto, Canada. Association for Computational Linguistics.

Alexandra N. Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72:1385–1470. Publisher Copyright: © 2021 AI Access Foundation. All rights reserved.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference.

Tharindu Cyril Weerasooriya, Alexander Ororbia, Raj Bhensadadia, Ashiqur KhudaBukhsh, and Christopher Homan. 2023. Disagreement matters: Preserving label diversity by jointly modeling item and annotator label distributions with DisCo. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4679–4695, Toronto, Canada. Association for Computational Linguistics.

Nicolas Zampieri, Carlos Ramisch, Irina Illina, and Dominique Fohr. 2022. Identification of multiword expressions in tweets for hate speech detection. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 202–210, Marseille, France. European Language Resources Association.

A Data

We retrieved the HateWiC dataset upon request which is available for research purposes, licensed under CC BY-NC 4.0.

A.1 Sense-level annotation

The annotation task was conducted on Wiktionary definitions from the HateWiC dataset, comprising nearly 1,900 instances. Each instance was annotated with two categorical labels: one for semantic Domain and one for Person Aspect. The Domain label captures the conceptual domain of the term, provided that its part of speech is a noun; otherwise, it is labeled as *NotNoun*. The Person Aspect label identifies what aspect of a person the sense pertains to, and is only applied if the term refers to a person; otherwise, it is labeled as *NotPerson*.

The Domain taxonomy includes the following categories: *Person*, *Animal*, *Artefact*, *Body part*, *Disease*, *Food*, *Plant*, *Supernatural being*, *Ambiguous* and *Other*. The Person Aspect taxonomy includes: *Personality/behavior*, *Ethnicity/nationality*, *Health/disability*, *Intelligence*, *Profession*, *Politics/ideology*, *Appearance*, *Gender/sexuality*, *Kinship/social*, *Social class*, *Age* and *Unspecified*.

Full annotation guidelines, including definitions of each category, are available in our GitHub repository. The annotation was carried out by two annotators with expertise in linguistics: Annotator 1 (author) is a PhD student in Computational Linguistics and Annotator 2 is a student in English and Computational Linguistics. Inter-annotator agreement, measured using Cohen's kappa, was $\kappa = 0.832$ for the Domain annotations and $\kappa = 0.764$ for the Person Aspect annotations. Annotator 2's annotations served as validation, with Annotator 1 providing the authoritative judgment when consensus was not reached.

A.2 Grammatical Role extraction

We implemented a custom pipeline using the `spacy` nlp library with the `en_core_web_sm` model. To locate predefined (multiword) terms within sentences, we used spacy's `PhraseMatcher`, configured to match on the lemmatized form of the target terms (using spacy's built-in lemmatizer). If no exact lemmatized match was found, approximate string matching was performed using the `rapidfuzz` library, leveraging the Levenshtein similarity ratio. Candidate noun chunks in each sentence were compared to the expected lemmatized term, and the highest-scoring match above a fuzzy similarity threshold of 85 was selected. For both exact and approximate matches, the syntactic role of the term was determined by extracting the dependency label (`dep_`) of the syntactic head of the matched span. Processing was parallelized using spacy's `nlp.pipe` API with a batch size of 50.

After extracting the dependency parsing tags using SpaCy for the target terms in the texts, we mapped them to a coarser categorization based on guidelines provided in https://github.com/clir/clearnlp-guidelines/blob/master/md/specifications/dependency_labels.md. The coarse-grained categories of Grammatical Roles are: *subject*, *object*, *nominal*, *adverbial*, *preposition*, *coordination*, *root*, *compoundword*, *complement* and *miscellaneous*.

A.3 Annotator Hatefulness Ratio

We computed each annotator's Hatefulness Ratio, defined as the proportion of instances they labeled as hateful across the dataset, i.e.:

$$H_a = \frac{N_a^{(h)}}{N_a}$$

where H_a denotes the Hatefulness Ratio of annotator a, $N_a^{(h)}$ is the number of instances annotator a labeled as hateful, and N_a is the total number of instances annotated by a.

B Empirical Analysis

Inter-annotator agreement was computed using the `krippendorff` package. Furthermore, we conducted two types of statistical analyses. Feature association testing was carried out using chi-squared tests of independence via the `scipy.stats` package. For handling numerical variables, we applied quantile-based binning to create discrete categories. This was achieved using the `qcut` function from the `pandas` library.

For the analysis visualized in Figure 4, ordinary least squares (OLS) regression was applied using the `OLS` method from the `statsmodels.api` module. We included interaction and polynomial terms using `PolynomialFeatures` from `sklearn.preprocessing` and computed the coefficient of determination (R^2) with `sklearn.metrics.r2_score`.

All data visualizations were produced with `matplotlib.pyplot` and `seaborn`.

C Computational Modeling

All modeling experiments were implemented using the PyTorch framework. Text representations were obtained using a pre-trained transformer model. More specifically, initialized with the `'answerdotai/ModernBERT-base'` checkpoint via the `transformers` library. Model training was performed with the Adam optimizer using a learning rate of 2×10^{-5} and a batch size of 32 for both training and evaluation. The training process was conducted over 3 epochs using a fixed random seed of 56 to ensure reproducibility. Classification performance was evaluated using cross-entropy loss and accuracy computed with `sklearn.metrics`. All experiments were executed on a single NVIDIA RTX A6000 GPU using CUDA acceleration.

D Additional Results

D.1 Main effect of Hatefulness Ratio

An additional illustration of the directions of feature effects is provided in Figure 4. The figure plots individual annotators' hateful labeling tendency (Hatefulness Ratio) against their annotation

agreement ratio with the majority, which allows for more concrete interpretation of this measured main effect as presented in Table 2. Unlike earlier analyses, which relied on binned categories, this figure presents the continuous relationship between these variables. The relationship appears weakly quadratic, with lower agreement visible at both extremes of Hatefulness Ratio. As expected, annotators who rarely or frequently label instances as hateful tend to deviate more often from the majority decision, while those with moderate Hatefulness Ratios agree more frequently. Especially for these annotators, incorporating individual labeling behavior may improve models of annotation variation.

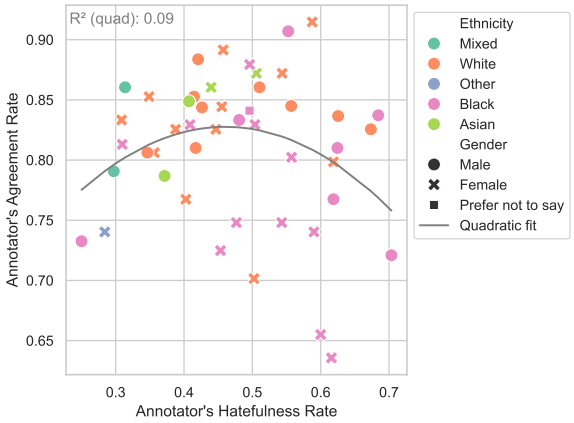


Figure 4: Annotator's hatefulness proportion (i.e. how much of their annotations is hateful) against agreement ratio (i.e. how much of their annotations is the majority vote). Each datapoint represents one annotator.

D.2 Contingency tables

For each feature, the frequency counts that underlie the statistical analyses in Table 2 are reported in Tables 4 until 13.

Person Aspect	agree	disagree
Age	77	16
Appearance	307	41
Ethnicity/nationality	480	80
Gender/sexuality	510	74
Health/disability	171	17
Intelligence	380	52
Kinship/social	164	14
NotPerson	4271	688
Personality/behavior	2183	473
Politics/ideology	772	185
Profession	228	43
Social class	49	10
Undecided	47	10
Unspecified	177	18

Table 4: Frequencies for individual annotations by Agreement with the majority vote and Person Aspect

Domain	agree	disagree
Ambiguous	191	28
Animal	248	40
Artefact	579	50
Body part	212	46
Disease	213	43
Food	110	15
NotNoun	2111	377
Other	802	120
Person	6786	1107
Plant	82	12
Super natural being	33	7

Table 5: Frequencies for individual annotations by Agreement with the majority vote and Domain

Context Length	agree	disagree
3-14	2667	341
14-23	2494	374
23-35	2478	512
35-176	2483	487

Table 6: Frequencies for individual annotations by Agreement with the majority vote and Context Length

Grammatical Role	agree	disagree
adverbial	457	55
complement	415	73
compoundword	704	125
coordination	687	125
miscellaneous	101	18
nominal	1078	188
not_found	196	48
object	2897	486
preposition	1073	373
root	600	91
subject	1246	297

Table 7: Frequencies for individual annotations by Agreement with the majority vote and Grammatical Role

POS	agree	disagree
adjective	848	159
adverb	237	30
interjection	47	5
noun	7688	1421
phrase	3	0
proper noun	94	11
verb	1193	210

Table 8: Frequencies for individual annotations by Agreement with the majority vote and POS

Hate Ratio	agree	disagree
0.25-0.4	2632	480
0.4-0.48	2668	482
0.48-0.56	2577	382
0.56-0.7	2418	535

Table 9: Frequencies for individual annotations by Agreement with the majority vote and Hate Ratio

Ethnicity	agree	disagree
Asian	889	142
Black	3883	810
Mixed	639	110
Other	191	66
White	4511	675

Table 10: Frequencies for individual annotations by Agreement with the majority vote and Ethnicity

Age Category	agree	disagree
20-29	3657	1255
30-39	2790	960
40-49	210	20
50-59	208	40
60+	213	32

Table 11: Frequencies for individual annotations by Agreement with the majority vote and Age Category

Gender	agree	disagree
Female	5410	1086
Male	4467	777
Prefer	217	27

Table 12: Frequencies for individual annotations by Agreement with the majority vote and Gender

Annotator Id	agree	disagree
annotator_1	261	33
annotator_10	213	32
annotator_13	181	32
annotator_14	213	32
annotator_16	198	59
annotator_18	193	39
annotator_19	187	60
annotator_2	228	24
annotator_22	215	43
annotator_23	217	27
annotator_24	220	26
annotator_25	216	31
annotator_26	169	31
annotator_28	219	29
annotator_30	217	29
annotator_31	222	17
annotator_34	215	41
annotator_35	207	37
annotator_36	191	53
annotator_37	226	28
annotator_39	193	42
annotator_4	206	51
annotator_42	214	38
annotator_44	222	35
annotator_47	191	66
annotator_5	225	27
annotator_53	208	36
annotator_55	164	78
annotator_56	198	18
annotator_58	225	28
annotator_59	220	38
annotator_6	189	39
annotator_60	222	33
annotator_62	234	23
annotator_63	186	67
annotator_64	209	44
annotator_65	209	26
annotator_66	230	28
annotator_69	213	40
annotator_71	213	32
annotator_74	214	44
annotator_75	203	50
annotator_77	215	36
annotator_78	208	49
annotator_79	218	37
annotator_8	216	35
annotator_83	204	45
annotator_85	236	21

Table 13: Frequencies for individual annotations by Agreement with the majority vote and Annotator Id

Context Effects on the Interpretation of Complement Coercion: A Comparative Study with Language Models in Norwegian

Matteo Radaelli
Norwegian University of Science and Technology
matteo.radaelli@ntnu.no

Emmanuele Chersoni
The Hong Kong Polytechnic University
emmanuele.chersoni@polyu.edu.hk

Alessandro Lenci
University of Pisa
alessandro.lenci@unipi.it

Giosuè Baggio
Norwegian University of Science and Technology
giosue.baggio@ntnu.no

Abstract

In complement coercion sentences, like *John began the book*, a covert event (e.g., reading) may be recovered based on lexical meanings, world knowledge, and context. We investigate how context influences coercion interpretation performance for 17 language models (LMs) in Norwegian, a low-resource language. Our new dataset contained isolated coercion sentences (context-neutral), plus the same sentences with a subject NP that suggests a particular covert event and sentences that have a similar effect but that precede or follow the coercion sentence. LMs generally benefit from contextual enrichment, but performance varies depending on the model. Models that struggled in context-neutral sentences showed greater improvements from contextual enrichment. Subject NPs and pre-coercion sentences had the largest effect in facilitating coercion interpretation.

1 Introduction

Coercion results from a semantic type mismatch between a predicate and its argument (Pustejovsky, 1991, 1995; Jackendoff, 1997). In *John began the book*, the aspectual verb *begin* requires an event-denoting complement, but is instead combined with an entity-denoting NP (*the book*). The covert event can be recovered by exploiting the meaning of lexical constituents, world knowledge, and context (Pustejovsky, 1991, 1995; Lapata and Lascarides, 2003). Hence, speakers can interpret the sentence above as meaning, for example, that John began *reading* the book. Because the resulting interpretation is not a strict function of constituent meanings and syntax, coercion has been argued to violate strong versions of the principle of compositionality (Asher, 2015; Jackendoff, 1997; Baggio et al., 2012, 2016). Experiments found longer reading times (McElree et al., 2001; Traxler et al., 2002) and on-line processing costs (Pylkkänen and McElree, 2007; Baggio et al., 2010; Baggio, 2018) for coercion sentences compared to controls in which the relevant event is expressed by a non-aspectual verb (e.g., *John read the book*) or an event-denoting complement.

Transformer-based language models (LMs) (Vaswani et al., 2017) have become popular in NLP owing to their success in a range of tasks. However, few studies addressed how LMs process complement coercion. Previous research focused mainly on coercion as a challenge for sentence interpretation and framed it as a task where LMs have to predict the best covert event given an aspectual verb–complement combination (Rambelli et al., 2020; Ye et al., 2022; Gietz and Beekhuizen, 2022; Im and Lee, 2024; Rambelli et al., 2024). Radaelli et al. (2025) demonstrate that LMs have difficulty retrieving covert events for coercion sentences in Norwegian in the absence of context. The present study extends that work by investigating the role of context. Transformers' self-attention mechanism captures local contextual information by assigning greater relevance to some tokens compared to others within a sequence (Vaswani et al., 2017; Devlin et al., 2019; Radford et al., 2019). The result is the generation of dynamic linguistic representations that vary according to the surrounding context. We expect that contextual information will improve the performance of transformer-based LMs in covert event interpretation of coercion sentences.

2 Theories of Coercion in Context

One hypothesis assumes that a coercion interpretation is the result of *enriched composition*: lexico-semantic properties of words are leveraged to enrich the meaning of the sentence, resulting in an eventive reading (Pustejovsky, 1991, 1995, 1998; Asher, 2015). Each lexical item is associated with a *qualia structure* that includes, among others, a specification of TELIC (the purpose of an object)

Proceedings of the 16th International Conference on Computational Semantics, pages 88–98
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

and AGENTIVE (how an object is created) roles of the relevant entity. For coercion sentences, a type mismatch between the aspectual verb and its complement leads to the recovery of the covert event by exploiting the qualia roles of the entity *book*: the TELIC role implies that *reading* is the covert activity, while the AGENTIVE role implies *writing*. Contextual information can motivate different interpretations than those suggested by default qualia roles (Pustejovsky, 1995; Pustejovsky and Bouillon, 1995; Pustejovsky, 1998; Traxler et al., 2002). In *The author began the book*, the subject NP can facilitate the recovery of the AGENTIVE quale *write* (Traxler et al., 2005). In *The climber enjoyed the rock*, instead, where no specific TELIC role is provided by *rock*, the complement is enriched through co-composition of the subject NP *climber*, suggesting the interpretation that the climber enjoyed climbing the rock (Pustejovsky, 1998, p. 294).

The contextual enrichment of coercion sentences is also motivated by empirical studies. McElree et al. (2001, p. 7), for instance, acknowledge that the "properties of the subject NP appear to determine the default interpretation in an otherwise neutral context." In eye-tracking experiments, Traxler et al. (2005) concluded that contextual information does not necessarily attenuate processing costs in coercion sentences, but can be exploited as an 'extended lexicon', licensing an eventive interpretation that could be otherwise costly to generate.

The pragmatic hypothesis proposes a different account of complement coercion compared to the more constrained approach of the lexical analysis, which claims that coercion sentences are enriched solely via default qualia-based lexical information (Lascarides and Copestake, 1998; Zarcone et al., 2014). Building on relevance theory (Sperber and Wilson, 1986; Falkum, 2015), the proposal by De Almeida (2004) and De Almeida and Dwivedi (2008) claims that lexical entries only specify an expression's denotation or type (Fodor and Lepore, 1998). The interpretation of coercion sentences is therefore not lexically-driven but guided by post-lexical pragmatic inferences, world knowledge, and context. This leads to more flexible interpretations and a wider variety of readings compared to those afforded by qualia roles (Fodor and Lepore, 1998; De Almeida, 2004; Falkum, 2015).

Experimental work by Zarcone and Padó (2011) and Zarcone et al. (2014) provides instead support for Generalized Event Knowledge (GEK) (McRae and Matsuki, 2009) in coercion interpretation, an alternative to both lexical qualia-based and pragmatic hypotheses. The words-as-cues hypothesis (Elman, 2009) claims that speakers store event knowledge in memory: words serve as cues that allow access to such knowledge, modulating expectations about upcoming words. In a self-paced reading study, Zarcone et al. (2014) found that if coercion interpretations align with more typical events, sentences are read faster.

According to Piñango and Deo (2016), aspectual verbs do not necessarily trigger coercion effects when combined with entity-denoting complements, but can also specify mereological (i.e., part-whole) relationships between arguments (e.g., *The perch begins the trail*) as well as causal relations between events. In this theory, aspectual verbs select *structured individuals*, with parts ordered along a particular axis (e.g., spatial, temporal, informational etc.), formally a 'one-dimensional directed path structure' (DPS). Each argument encodes a set of functions that guide the mapping relative to a specific dimension. Both stative and eventive readings for sentences with aspectual verbs are possible. In the aspectually stative sentence *A thunderstorm began the day*, the predicate specifies the existence of a part-whole relation between the denotata of the complement and the subject. The information provided by the complement allows the predicate function to map the arguments onto a temporal dimension, interpreting the subject *thunderstorm* – a non-agentive entity – as denoting the initial temporal sub-interval of the denotation of the complement *morning* (Piñango and Deo, 2016, p. 369).

In sentences like *John began the book*, Piñango and Deo (2016) argue that the aspectual verb does not impose any type-selectional restrictions, hence no type-mismatch repair is needed. They propose instead a 'structured mapping' via inverse thematic functions. Because the event is underspecified, the traditional thematic function, which relates events to their participants, is not available. The inverse thematic function allows mapping of "pairs of individuals to the smallest event that the individual bears a participant role to at that time in a given context" (p. 385). Argument denotations and sentence context provide further constraints on the recovery of the event. Since complements are semantically undetermined and can map onto several possible axes, the same sentence can also be interpreted statively. If *John* is not interpreted as an agent, the

arguments would be mapped onto an informational dimension instead of an eventive one, and John would be considered one subpart of the informational object *the book*: John's work would then be an initial part of the book, such as a first chapter.

3 LMs and Complement Coercion

The first study evaluating LMs on complement coercion was Rambelli et al. (2020), who analyzed the performance of pretrained Transformers of the BERT and GPT families. They used datasets from different behavioral studies (McElree et al., 2001; Traxler et al., 2002; Lapata and Lascarides, 2003). The results revealed that Transformer-based models behaved differently from each other depending on the model's framework. ROBERTA, for example, emerged as the most robust LM, performing better than other models on the Lapata-Lascarides dataset (Lapata and Lascarides, 2003), with 80% accuracy in binary classification and 73% in a correlation task. In contrast, GPT-2 appeared to be more unstable, with a better score in the binary classification task (87%) but poorer performance in the correlation task (43%). Vanilla BERT, on the other hand, showed a marginal improvement over the baseline, suggesting a limited ability for contextualized embeddings in capturing eventive information from context. Finally, the authors report that distributional and non-Transformer frameworks, such as the Structured Distributional Model (SDM), performed similarly to ROBERTA despite being pretrained on smaller datasets.

Gietz and Beekhuizen (2022) consider coercion as a case of flexible semantic enrichment based on context, rather than as obligatory semantic completion. They analyzed a vanilla BERT model using a dataset with naturally-occurring coercion sentences from the COCA Corpus, successively annotated by humans. They argue that traditional 'hand-crafted' coercion sentences from previous studies always allow clear event interpretations, while naturally-occurring sentences usually include additional contextual information. BERT performed well in cases where consensus between annotators on a covert event was high, but struggled with sentences with less consensus. The model benefited from contextual information, improving event prediction.

Ye et al. (2022) used a dataset of naturally-occurring coercion sentences extracted from the C4 Corpus (Raffel et al., 2020). The authors argue that the process of coercion interpretation is analogous to paraphrasing: the coercion sentence is rephrased in a way that ambiguity is eliminated and the covert event is revealed. They found that pretrained BERT has difficulty with coercion interpretation, while a model fine-tuned with explicitly paraphrased sentences leads to better performance.

Radaelli et al. (2025) investigate whether LMs can leverage syntactic structure and lexical meaning toward recovering covert events. They conduct a large-scale evaluation of LMs in Norwegian, a low-resource language with variable grammatical realization of coercion, which partly depends on the aspectual verb used. Initiation verbs usually combine with entity-denoting NPs in PPs introduced either by *på* or by *med* (*John begynte på/med boken*; 'John began on/with the book'). With continuation and cessation verbs, complements are mainly introduced by *med*-PPs or directly an NP (*John avluttet med/ø boken*, 'John finished (with) the book'). Radaelli et al. (2025) released a new dataset of sentence pairs, each containing a context-neutral coercion sentence and an event resolution prompt. The dataset included 90 distinct entities from 6 different categories, and the syntactic realization of coercion was varied systematically by the aspectual verb and PP/NP. The study tested 17 Norwegian LMs, spanning BERT-like autoencoders and autoregressive models. In general, LMs struggled to recover implicit events. Surprisal estimates for whole sentences indicate that most LMs tested are unable to leverage the syntactic structure of the VP to interpret coercion items, showing no significant performance changes across syntactic constructions. For more details, see Section 5.1.

4 Task Proposal

Here, we explore the role of context in coercion interpretation, extending Radaelli et al. (2025)'s work on Norwegian context-neutral sentences. We study how different types of context influence the prediction distributions for LMs in a covert event interpretation task. We used the same evaluation strategy as Radaelli et al. (2025): instead of assessing models' performance only on a pre-defined set of top-1 ranked predictions as gold standard, we considered the *ranked prediction distribution* for each model; for each coercion sentence, a model must output a set of top-5 ranked predictions $O = o_1, ..., o_5$.

The distribution is then evaluated by calculating the mean average precision metric, which captures the consistency of LMs in predicting appropriate

events (see below) in the top ranking. We consider a model 'sensitive' to coercion, if it can provide a prediction distribution that is relevant to event interpretations: given a coercion sentence, expressed as a triplet ⟨*subject, aspectual verb, entity*⟩, we expect a redistribution of output predictions in a way that eventive interpretations are at the top of the ranking. The addition of contextual information should lead to further redistribution of the outputs, possibly with a shift towards the event interpretations suggested by the context.

The output predictions for each sentence will be evaluated by considering any event (verb) as correct as long as it satisfies the semantic constraints required by coercion and by the context. Following Piñango and Deo (2016) and Spalek and Sæbø (2019), a covert event is a plausible candidate for coercion when its combination with subject and complement expresses telicity, implying a "natural endpoint or goal state" that is coherent with the overall meaning of the sentence. The class of *accomplishments* is our ground truth for event classification, as it specifies durative, dynamic, and telic situation types or Aktionsart (Vendler, 1967; Spalek and Sæbø, 2019). All predicted events that are accomplishments are compositionally appropriate candidates, including those that may be weakly associated in coercion contexts. For example, the triplet ⟨*goat, begin, book*⟩ can suggest the covert event *eat* (Lascarides and Copestake, 1998). Some events must however be discarded: although they belong to the accomplishment class, their combination with the given subject and object results in a semantic anomaly. For example, a verb like *klatre* ('climb') could be plausible when predicted with objects that afford movement (e.g., *mur*; 'wall') but not with food items (e.g., *salat*; 'salad').

4.1 Dataset

We adopted a dataset originally created by Radaelli et al. (2025). Each item is a sentence pair designed to elicit the generation of covert events. Each pair includes (1) a context-neutral coercion sentence:

(1) {SUBJ} {VERB-FIN} {PREP|Ø} {ENTITY-DEF}.
E.g.: *Kim begynte på boken.* ('Kim began the book')

and (2) a sentence that prompts event retrieval:

(2) Det som {SUBJ} {VERB-FIN} å gjøre, var å [MASK].
'What {SUBJ} {VERB-FIN} to do was to [MASK]'.

The sentences contained the following elements:

- A single gender-neutral proper name (*Kim*) as subject {*SUBJ*}.
- 90 complement entity-denoting definite nouns {*ENTITY-DEF*}, consisting of real artifacts as incremental theme arguments of the implicit event. These entities belong to six different semantic categories: *food*, *text*, *clothing*, *everyday objects* (e.g., bag), *construction/housing* (e.g., wall), and *entertainment* (e.g., graffiti).
- Four aspectual verbs {*VERB-FIN*} in simple past form (*preteritum*), i.e., ***begynne*** (begin), ***starte*** (start), ***fortsette*** (continue), and ***avslutte*** (finish). Aspectual verbs, in contrast to other classes like psychological verbs (e.g., enjoy), were considered the only class of verbs that robustly trigger complement coercion, as shown experimentally by Katsika et al. (2012).
- Three complement syntactic constructions {*PREP—Ø*} introduced by a PP with either *på* or *med* or directly by an NP.
- The masked token [MASK] is included only for autoencoder models. With autoregressive models, [MASK] is replaced by blank tokens, used to prompt the prediction of the next sentence token.

We extended this dataset, here condition (a), by introducing three new conditions (b-d), each providing controlled contextual information in a specific portion of the experimental item (Table 1). The contextual enrichment applies only to sentence (1) in each pair, leaving (2) unchanged:

(a) Context-neutral: as in the original dataset;

(b) Subject-enriched context: the neutral subject (*Kim*) is replaced with a subject NP relevant for particular covert events;

(c) Post-verbal context: additional text is added after the entity complement as an adjunct or a coordinated phrase;

(d) Pre-coercion sentence: a sentence is concatenated before the coercion sentence, providing a discourse-level context.

All items in (1) included sentences with similar token length, with length variation of 2-3 tokens. Subjects and entity NPs were always in definite form,

(1) Coercion Sentence	(2) Prompt Sentence for Event Interpretation
(a) Kim begynte på essayet. (b) **Tolken** begynte på essayet. (c) Kim begynte på essayet **ved hjelp av ordboken**. (d) **Kim ønsket å publisere sitt nye verk på et annet språk for en fransk avis**. Kim begynte på essayet.	Det som Kim/tolken begynte å gjøre, var å ([MASK]).

Table 1: Examples of coercion sentences with the aspectual verb *å begynne* (to begin) in context conditions (a–d) in Norwegian and a common event-prompt interpretation sentence. Contextual information is presented in bold. Translations into English: (1a) 'Kim began the essay', (1b) 'The interpreter began the essay', (1c) 'Kim began the essay with the help of the dictionary', (1d) 'Kim wanted to publish his new work in a different language for a French newspaper. Kim began the essay', (2) 'What Kim/the interpreter began to do was to ([MASK])'.

while aspectual verbs were in *preteritum* form (past simple). The context was always coherent with the verb-complement combination.

For the assessment of models' performance we compared the results by Radaelli et al. (2025) with context-enriched conditions. The extended dataset includes a total of 4320 sentence pairs in standard written Norwegian Bokmål.

4.2 Tested Models

We tested the extended dataset on 17 pretrained Norwegian LMs, with autoencoders, such as BERT (Devlin et al., 2019), and autoregressive models, such as GPT-2 (Radford et al., 2019), LLAMA-2 (Touvron et al., 2023), BLOOM (Scao et al., 2022), and MISTRAL (Jiang et al., 2023). Table 2 shows the list of the language models tested here. The models differ considerably not only in architecture, but also in number of parameters and size of training data. Most LMs tested are monolingual models, only two (MBERT-CASED/UNCASED) are multilingual, while NORMISTRAL-7B-WARM was primarily pretrained in English and further trained in Norwegian. All tested models are available on Hugginface.[1]

4.3 Baseline Model

To assess performance between models and between different contextual conditions, we leveraged the same statistical baseline model as Radaelli et al. (2025): plausibility of event estimates were based on Pointwise Mutual Information (PMI) (Church and Hanks, 1990) between the verb and its object. The result is a list of (eventive) verbs strongly associated with an entity. These estimates are based on the Norwegian Colossal Corpus (NCC) (Kummervold et al., 2022), an open source corpus employed for training most current Norwegian LMs.

[1]https://huggingface.co/

Model	# Par.	Tr. Data
MBERT CASED/UNCASED	178M	3.3B*
NB-BERT-BASE	178M	7B
NB-BERT	355M	7B
NORBERT	111M	1.9B
NORBERT2	125M	15B
NORBERT3-BASE	123M	25B
NORBERT3-LARGE	353M	25B
NORBERT3-SMALL	40M	25B
NORBERT3-XS	15M	25B
NORBLOOM-7B-SCRATCH	7B	26.7B
NORGPT-369M	369M	25B
NORGPT-3B	3B	25B
NORGPT-3B-CONTINUE	3B	25B
NORLLAMA-3B	3B	26.7B
NORMISTRAL-7B-SCRATCH	7B	26.7B
NORMISTRAL-7B-WARM	7B	26.7B

Table 2: Tested LMs with number of parameters (*#Par.*) and training data (*Tr. Data*). *The amount of training data for MBERT is shared over 114 different languages.

4.4 Performance Evaluation

All prediction outputs provided by a given LM were manually classified by two of the authors according to Aktionsart, assessing the plausibility of the prediction in the coercion sentence. Disagreements were resolved through discussion. Predictions that were grammatically irrelevant to coercion sentences were discarded. We adopted two evaluation metrics for assessing models' performance. The first is mean average precision (mAP), which evaluates the ranking quality of a specific model based on the weighted means of average precision scores (AP) in the set of all sentences (S) (Manning et al., 2009; Kotlerman et al., 2010):

$$\text{mAP} = \frac{1}{S}\sum_{s=1}^{S}\text{AP}(s)$$

For any given sentence s, the AP score takes into account the ranking of the top-5 output predictions:

$$\text{AP(s)} = \sum_{k=1}^{5} P(k)\cdot\Delta R(k)$$

where $P(k)$ is the precision score at rank k and $\Delta R(k)$ is the recall difference between the current k and its antecedent $k-1$. A high mAP score indicates that the model tends to predict and rank accomplishments at the top. A low mAP score suggests either that the model proposes an event from an Aktionsart class other than accomplishments, or that the predicted accomplishment is ranked lower. The second metric is the mean top-ranked accuracy (A1) across the entire set of sentences (S). In this case, for each sentence, only the top-ranked prediction will be considered. Similar to the previous score, accomplishments count as the correct outputs, while other classes are false positives.

5 Results

5.1 General Results

Radaelli et al. (2025) found that LMs generally struggle to identify plausible events in context-neutral coercion sentences: mAP and A1 scores were low across LMs. Only few models exceeded the statistical baseline, and their performance varied mainly by model architecture and size. BERT-like models performed better than autoregressive models, with NORBERT3 showing relatively strong performance. Among autoregressive models, only NORLLAMA-3B and NORMISTRAL-7B-WARM exceeded the baseline. Model size also played a role: only the larger NORBERT3 variants could reach higher results, and autoregressive LMs like NORLLAMA-3B also showed decent performance, most likely due to their size.

Table 3 shows the mAP and A1 scores of all LMs tested on the covert event interpretation task in Norwegian. For comparison, we included the context-neutral scores from Radaelli et al. (2025). The results are available on GitHub. On the mAP scores, contextual information generally improved performance for most models compared to context-neutral sentences: 9 models outperformed the baseline, compared to only 4 with coercion-neutral sentences. However, even with context, the remaining 8 models still showed difficulties in consistently predicting appropriate events. Contextual information appears to particularly improve prediction for autoencoder models. Most models in the NORBERT family performed relatively well, reaching mAP scores above the baseline. Smaller models like NORBERT3-BASE, NORBERT3-SMALL, and NORBERT2, which showed poor performance in context-neutral sentences, here outperformed even the best model NORLLAMA-3B in the context-neutral condition. NORBERT and NORBERT3-XS, on the other hand, still struggled with the task. Contextual information also improved performance of the NB-BERT family, namely LMs trained entirely on the NCC corpus, also used to create the statistical baseline model. While NB-BERT-LARGE achieved results above the baseline, NB-BERT-BASE still showed low performance despite the improvement.

Model	mAP			A1		
	No Ctx	W/Ctx	Diff	No Ctx	W/Ctx	Diff
NCC (Baseline)	0.59	0.59	0.00	0.47	0.47	0.00
MBERT-CASED	0.07	0.07	0.00	0.00	0.01	0.01
MBERT-UNCASED	0.27	0.36	0.09	0.22	0.32	0.10
NORGPT-369M	0.56	0.62	0.06	0.54	0.57	0.03
NORGPT-3B	0.48	0.62	0.14	0.42	0.55	0.13
NORGPT-3B-CONT.	0.46	0.58	0.13	0.42	0.50	0.08
NORLLAMA-3B	0.71	0.66	-0.06	0.67	0.61	-0.06
NB-BERT-BASE	0.38	0.57	0.19	0.33	0.49	0.16
NB-BERT-LARGE	0.54	0.67	0.13	0.47	0.61	0.14
NORBERT	0.25	0.36	0.11	0.18	0.30	0.12
NORBERT2	0.44	0.69	0.24	0.34	0.62	0.28
NORBERT3-BASE	0.63	0.73	0.11	0.58	0.69	0.11
NORBERT3-LARGE	0.60	0.65	0.05	0.55	0.56	0.01
NORBERT3-SMALL	0.59	0.73	0.14	0.55	0.69	0.14
NORBERT3-XS	0.29	0.43	0.14	0.16	0.30	0.14
NORBLOOM-7B-S.	0.46	0.56	0.10	0.34	0.45	0.11
NORMISTRAL-7B-S.	0.38	0.58	0.19	0.29	0.49	0.20
NORMISTRAL-7B-W	0.63	0.64	0.01	0.54	0.56	0.02

Table 3: Comparison of mean average precision (mAP) and mean top-ranked accuracy (A1) for covert event retrieval in Norwegian context-neutral (No Ctx) and context-enriched (W/Ctx) sentences. Results for No Ctx are provided by Radaelli et al. (2025).

A different pattern is found for autoregressive models. Most GPT-2 models still struggled to perform at or above the baseline. Only NORGPT-369M and NORGPT-3B benefited from the context, reaching reasonable results in mAP scores. NORBLOOM-7B-SCRATCH and NORMISTRAL-7B-SCRATCH still showed poor performance despite contextual enrichment, remaining below the baseline, while NORMISTRAL-7B-WARM did not improve relative to context-neutral sentences. Finally, NORLLAMA-3B is the only model that apparently suffers from the presence of context, showing a performance drop.

Analyzing the difference of mAP scores in sentences with and without context, we can appreciate how much context-enriched sentences enhanced the models' performance. First, context generally increases performance for most of those LMs that in the context-neutral condition struggled with coercion resolution. For example, NORBERT2, NB-BERT-BASE, and NORMISTRAL-7B-SCRATCH showed a significant improvement. MISTRAL and BERT-like models demonstrate the ability to exploit

context more effectively to improve performance while they struggled in the context-neutral condition, regardless of parameter sizes. GPT models also showed positive but weaker improvements, especially those with higher parameter sizes, such as NORGPT-3B and NORGPT-3B-CONTINUE. On the other hand, models that previously obtained relatively high mAP scores either did not show a significant change (e.g., NORMISTRAL-7B-WARM) or performed worse (e.g., NORLLAMA-3B).

A similar trend emerges from an analysis of A1 scores. NORBERT3-SMALL and NORBERT3-BASE reached the highest A1 score, close to 0.70. The other models showed considerably lower performance. Even the 10 models that outperformed the baseline obtained an A1 score ranging from 49 to 62, indicating that models still fail to top-rank accomplishments in approximately half of the cases.

A qualitative error analysis revealed that the addition of contextual information can sensibly affect model's performance. For example, comparing the subject-enriched sentence *Fienden begynte med testamentet* ('The enemy began with the will') to its neutral counterpart (*Kim begynte med testamentet*) on NORBERT3-BASE, we observed differences in the ranking. In the context-neutral case, the top-5 predictions were ⟨*skrive* ('write'), *lage* ('make'), *ta* ('take'), *gjøre* ('do'), *bruke* ('use')⟩, with the first two events being the only plausible accomplishments for coercion interpretation. In the context-enriched cases, the model kept the accomplishment (*skrive*) but prioritized verbs like *drepe* ('kill') and *stjele* ('steal'), indicating subject-driven biases. This means that, in this case, the replacement with a subject NP enriched with additional semantic information strongly shifts the prediction space of the model to events that are closely related to it. However, despite coherence with the subject, such outputs cannot be accepted: *drepe* requires an animate patient, while *stjele* lacks the durativity typical of accomplishments. Such events do not consider the contextual information conveyed by entire sentences, in particular the combination verb-entity. This suggests that contextual cues, especially those provided by the subject may strongly override the prediction ranking, guiding the model to predictions associated with those cues rather than by a compositional requirements.

Radaelli et al. (2025) conducted a quantitative error analysis with focus on the best performing model NORLLAMA-3B, examining the general frequency distribution of the predicted verbs across all context-neutral coercion sentences in the experiment. The analysis showed that the model produced a limited set of 68 unique verbs over 5,400 predictions, with the most frequent ones denoting either particularly generic events (e.g., *lage*, make, which combines with a wide range of entitites) or non-accomplishment verbs, here considered as false positives. In this study, we adopted the same analysis approach, by inspecting NORBERT3-BASE and comparing the event distribution across context-neutral and context-enriched coercion sentences. Table 4 shows the distribution of the first 15 most predicted events in all coercion sentences, comparing both context-neutral and context-enriched sentence conditions. The results suggest a similar trend to that found by Radaelli et al. (2025). First, also this model predicted a limited set of unique events, from 50 with context-neutral coercion sentences (among 5,400 predictions made in 1,080 sentences) increasing to 93 in context-enriched sentences (16,200 predictions in 3,240 sentences), suggesting that the addition of contextual information increases the variability of predicted events. Second, the ranking of predictions in both conditions is similar, following a skewed Zipfian distribution: the top ranked verbs dominate the distribution (covering up to 15% of the entire verb set), whereas predictions at lower positions show a sharp decrease of frequency. Finally, this analysis shows a minimal ranking variation in the distribution of verbs across

Verb	**No Ctx (Rel. Freq)**	**W/Ctx (Rel. Freq.)**
spille (play)	803 (0.15)	1,493 (0.09)
skrive (write)	781 (0.14)	1,924 (0.12)
le (laugh)	630 (0.12)	1,251 (0.08)
telle (count)	577 (0.11)	1,317 (0.08)
slå (hit)	524 (0.10)	1,455 (0.09)
danse (dance)	438 (0.08)	623 (0.04)
regne (calculate/rain)	414 (0.08)	1,117 (0.07)
vente (wait)	398 (0.07)	1,871 (0.12)
male (paint)	260 (0.05)	1,471 (0.09)
gå (go)	72 (0.01)	237 (0.01)
tale (speak)	65 (0.01)	392 (0.02)
lage (make)	56 (0.01)	644 (0.04)
holde (hold)	48 (0.01)	- (-)
rape (burp)	40 (0.01)	- (-)
bli (become / stay)	35 (0.01)	- (-)
sy (sew)	- (-)	316 (0.02)
hjelpe (help)	- (-)	233 (0.01)
bygge (build)	- (-)	185 (0.01)

Table 4: Top 15 events predicted by NORBERT3-SMALL across context-neutral (No Ctx) and context-enriched (W/Ctx) sentences, including both absolute and relative frequencies.

the two conditions, suggesting that context could not effectively elevate accomplishment verbs to the top rank, but influenced primarily the lower positions (e.g., *sy*, sew). Moreover, the most predicted events are in both conditions non-accomplishments, and therefore false positives for the classification task, usually denoting generic events not directly related to coercion resolution.

5.2 Context Types

We conducted further analyses of the impact of different context types on coercion sentences, with the conditions outlined in Section 4.1. For simplicity, we will consider only four models for this analysis: NORBERT3-SMALL, one of the top-performing models in this experiment, NORBERT2, which showed clear improvements compared to the context-neutral results by Radaelli et al. (2025), NORGPT-3B, the best performing GPT-based model, and NORLLAMA-3B, that showed instead a performance drop. Table 5 shows the models' mAP and A1 scores according to context conditions (b-d), including the context-neutral scores from Radaelli et al. (2025) as condition (a).

All context types led to improvements, with varying scores across conditions and LMs. Condition (d), the pre-coercion sentence, improved performance most, followed by condition (b), the context-enriched subject. Post-verbal context in condition (c) contributed the least among all conditions. A closer look at the scores reveals performance differences between LMs. First, NORBERT2 appears to benefit most when we consider the percentage increase over the mAP and A1 scores under contexts (b) and (d), with around 69% and over 90% improvement respectively compared to condition (a). This gap between the scores suggests that the model changed drastically the prediction distribution of verbs, ranking accomplishments at the top.

A more moderate performance improvement is found for both NORBERT3-SMALL and NORGPT-3B, which showed a similar behavior. On the one hand, their relative change against the baseline is small compared to NORBERT2, with a range between 25-33% for mAP scores and 28-39% for A1 scores. In this case, the gap between the mAP and A1 scores is minimal, meaning that the prediction distribution was more stable.

Finally, compared to the other models under test, NORLLAMA-3B shows the opposite trend in performance. Condition (c), the one that contributed least to the improvement, here leads to the largest decrease in performance, while a minimal positive improvement is observed for condition (b). Its A1 scores, however, remained unchanged under all conditions, showing only a minimal decrease.

Model	Cond.	mAP	A1
NORLLAMA-3B	a	0.713	0.670
NORLLAMA-3B	b	0.717 (0.004)	0.665 (-0.005)
NORLLAMA-3B	c	0.601 (-0.111)	0.547 (-0.123)
NORLLAMA-3B	d	0.653 (-0.060)	0.605 (-0.065)
NORBERT2	a	0.444	0.338
NORBERT2	b	0.718 (0.274)	0.650 (0.312)
NORBERT2	c	0.587 (0.143)	0.511 (0.173)
NORBERT2	d	0.753 (0.309)	0.708 (0.370)
NORBERT3-SMALL	a	0.593	0.545
NORBERT3-SMALL	b	0.747 (0.154)	0.699 (0.154)
NORBERT3-SMALL	c	0.676 (0.083)	0.608 (0.063)
NORBERT3-SMALL	d	0.776 (0.183)	0.762 (0.217)
NORGPT-3B	a	0.478	0.418
NORGPT-3B	b	0.625 (0.148)	0.536 (0.118)
NORGPT-3B	c	0.592 (0.115)	0.534 (0.116)
NORGPT-3B	d	0.639 (0.161)	0.571 (0.153)

Table 5: Mean average precision (mAP) and mean top-ranked accuracy (A1) for covert event retrieval in Norwegian across context-neutral (a) and context-enriched conditions (b-d). The results for (a) are from Radaelli et al. (2025).

6 General Discussion and Conclusion

Our results indicate that contextual enrichment of coercion sentences in Norwegian generally leads to better prediction distributions of covert events in almost all tested LMs. Additional context in specific sentence regions, such as in subject position, or the inclusion of sentences preceding the coercion construction, leads to most benefits in performance.

In this study, we found that performance varies also according to LM framework: BERT-like autoencoders appear to benefit most from contextual enrichment as compared to autoregressive models. This is consistent with the conclusion of Radaelli et al. (2025), where LMs were tested on coercion sentences without context. The advantage for autoencoders may be their bidirectional self-attention mechanism, which may be better able to capture semantic relations between constituents. However, models such as MBERT, NB-BERT-BASE, and NORBERT3-XS, for example, still showed only marginal improvements when exposed to context. Better performance for such models may be related to the interplay between their size and the amount of pretraining data: the multilingual model was one of the worst performing probably due to

its small training data in Norwegian. Conversely, results from NORBERT3-XS, suggest that, despite the large pretraining data, a smaller model still has limitations. Performance increases when the model's size increases, as shown for the larger NORBERT3 models. Other factors could also play a role. The NORBERT family showed more robust performance compared to NB-BERT models and MBERT, probably because the model was trained entirely from scratch on Norwegian and employed a custom WordPiece vocabulary. In contrast, NB-BERT starts from the MBERT framework and is trained on additional data in Norwegian without further changes (Kutuzov et al., 2021). Moreover, the third NORBERT generation, also introduces optimized training methods by excluding the next sentence prediction task and improving the masked language modeling objective task, increasing the span-based masking rather than masking single tokens (Samuel et al., 2023).

From the analysis of LM scores, we also found a consistent pattern linking their performance on context-neutral sentences and their improvement when context is introduced. Specifically, models that previously obtained poor results appear to benefit the most from context. Models like NORBERT2, NB-BERT-BASE, and NORBERT3-XS obtained a significant boost in performance compared to others. Such an improvement is however relative to their poor performance in context-neutral sentences. Their capacity to exploit contextual signals appears to compensate for such limitations.

It is particularly noteworthy that the LMs that obtained relatively high scores with context-neutral items are those that also showed more limited improvement when context is provided. This claim requires further research, but we hypothesize that such behavior may reflect a form of 'encoding saturation' by Transformer-based models, manifested in a limited capacity to integrate additional semantic information once a certain level of encoding complexity in a model's embedding-based representations has been reached. This behavior can also be observed when comparing models with almost identical architectures: NORBERT3-XS and NORBERT3-LARGE differ only in their parameter sizes, but they showed different improvement trends. We hypothesize that contextual information can compensate for gaps in world knowledge as required by coercion resolution. Consequently, context may not generally boost performance but rather benefits most the weaker models: stronger models show little change in their performance as they may have already reached a performance plateau, which cannot be improved by the integration of additional contextual information. This hypothesis is partially confirmed by the general improvement trend in results observed in Table 3.

Although contextual information generally led to better performance, LMs still show difficulties in interpreting complement coercion sentences. This aligns with the conclusions of earlier studies, such as Rambelli et al. (2020) and Ye et al. (2022) in English. It has been often observed that LMs lack a capacity for common sense reasoning based on plausible world models. This would also apply to natural language interpretation, in that current LMs have limited *linguistic common sense*: they lack the capacity to retrieve and exploit the kind of linguistic and world knowledge that would allow them to reliably make sense of complex, underspecified inputs (Lascarides and Copestake, 1998; Piñango and Deo, 2016; Baggio, 2018; Rambelli et al., 2024).

A closer look at our results sheds light on how and to what extent the behavior of Transformer models aligns with expectations based on different theoretical accounts on complement coercion. At first glance, the improvements seen for most models appear compatible with the pragmatic hypothesis: context and world knowledge can modulate or restrict coercion interpretations according to information that is not necessarily available from constituent meanings. However, such improvements were only seen for those models that were shown to be weaker in context-neutral scenarios, presumably due to a limited encoding of semantics in the learned embeddings. More semantically robust LMs were less influenced by context, suggesting that at least some relevant event information is encoded in the embeddings: this is more consistent with the lexical and Generalized Event Knowledge (GEK) hypotheses than with pragmatic accounts. On the other hand, our results cannot confirm the lexical hypothesis either, as context still has an effect in changing prediction distributions. Moreover, if models had learned and used lexically-bound representations such as qualia, we would not expect to see as outputs events that belong to incorrect Aktionsart, as in the example above. In addition, high performing models like NORLLAMA were even negatively influenced, suggesting a complex role of context in this task.

Acknowledgments

EC was supported by a GRF grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 15612222). We thank the anonymous reviewers for their valuable suggestions.

References

Nicholas Asher. 2015. Types, meanings and coercions in lexical semantics. *Lingua*, 157:66–82.

Giosuè Baggio, Travis Choma, Michiel Van Lambalgen, and Peter Hagoort. 2010. Coercion and compositionality. *Journal of Cognitive Neuroscience*, 22(9):2131–2140.

Giosuè Baggio, Keith Stenning, and Michiel Van Lambalgen. 2016. Semantics and cognition. In Maria Aloni and Paul Dekker, editors, *The Cambridge Handbook of Formal Semantics*, pages 756–774. Cambridge University Press.

Giosuè Baggio, Michiel Van Lambalgen, and Peter Hagoort. 2012. The processing consequences of compositionality. In Markus Werning, Wolfram Hinzen, and Edouard Machery, editors, *The Oxford Handbook of Compositionality*, pages 655–672. Oxford University Press, Oxford.

Giosuè Baggio. 2018. *Meaning in the Brain*. MIT Press.

Kenneth Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Roberto G De Almeida. 2004. The effect of context on the processing of type-shifting verbs. *Brain and language*, 90(1-3):249–261.

Roberto G De Almeida and Veena D Dwivedi. 2008. Coercion without lexical decomposition: Type-shifting effects revisited. *Canadian Journal of Linguistics/Revue canadienne de linguistique*, 53(2-3):301–326.

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*, pages 4171–4186.

Jeffrey L Elman. 2009. On the meaning of words and dinosaur bones: Lexical knowledge without a lexicon. *Cognitive Science*, 33(4):547–582.

Ingrid Lossius Falkum. 2015. The how and why of polysemy: A pragmatic account. *Lingua*, 157:83–99.

Jerry A Fodor and Ernie Lepore. 1998. The emptiness of the lexicon: Reflections on James Pustejovsky's The Generative Lexicon. *Linguistic Inquiry*, 29(2):269–288.

Frederick G Gietz and Barend Beekhuizen. 2022. Remodelling complement coercion interpretation. *Society for Computation in Linguistics*, 5(1).

Seohyun Im and Chungmin Lee. 2024. What gpt-4 knows about aspectual coercion: Focused on "begin the book". In *Proceedings of the Workshop on Cognitive Aspects of the Lexicon@ LREC-COLING 2024*, pages 56–67.

Ray Jackendoff. 1997. *The Architecture of the Language Faculty*. MIT Press.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Argyro Katsika, David Braze, Ashwini Deo, and Maria Mercedes Piñango. 2012. Complement coercion: Distinguishing between type-shifting and pragmatic inferencing. *The Mental Lexicon*, 7(1):58–76.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359–389.

Per Kummervold, Freddy Wetjen, and Javier de la Rosa. 2022. The Norwegian Colossal Corpus: A Text Corpus for Training Large Norwegian Language Models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3852–3860, Marseille, France. European Language Resources Association.

Andrey Kutuzov, Jeremy Barnes, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2021. Large-Scale Contextualised Language Modelling for Norwegian. ArXiv:2104.06546 [cs].

Maria Lapata and Alex Lascarides. 2003. A Probabilistic Account of Logical Metonymy. *Computational Linguistics*, 29(2):261–315.

Alex Lascarides and Ann Copestake. 1998. Pragmatics and word meaning. *Journal of Linguistics*, 34(2):387–414.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2009. *An Introduction to Information Retrieval*. Cambridge University Press, Cambridge.

Brian McElree, Matthew J Traxler, Martin J Pickering, Rachel E Seely, and Ray Jackendoff. 2001. Reading time evidence for enriched composition. *Cognition*, 78(1):B17–B25.

Ken McRae and Kazunaga Matsuki. 2009. People use their knowledge of common events to understand language, and do so as quickly as possible. *Language and Linguistics Compass*, 3(6):1417–1429.

Maria Mercedes Piñango and Ashwini Deo. 2016. Reanalyzing the Complement Coercion Effect through a Generalized Lexical Semantics for Aspectual Verbs. *Journal of Semantics*, 33(2):359–408.

James Pustejovsky. 1991. The Generative Lexicon. *Computational Linguistics*, 17(4):409–441.

James Pustejovsky. 1995. *The Generative Lexicon*. MIT Press.

James Pustejovsky. 1998. Generativity and explanation in semantics: A reply to Fodor and Lepore. *Linguistic Inquiry*, 29(2):289–311.

James Pustejovsky and Pierrette Bouillon. 1995. Aspectual Coercion and Logical Polysemy. *Journal of Semantics*, 12(2):133–162.

Liina Pylkkänen and Brian McElree. 2007. An MEG study of silent meaning. *Journal of Cognitive Neuroscience*, 19(11):1905–1921.

Matteo Radaelli, Emmanuele Chersoni, Alessandro Lenci, and Giosuè Baggio. 2025. Compositionality and Event Retrieval in Complement Coercion: A Study of Language Models in a Low-resource Setting. In *Proceedings of the 29th Conference on Computational Natural Language Learning (CoNLL)*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 1(8):9.

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.

Giulia Rambelli, Emmanuele Chersoni, Alessandro Lenci, Philippe Blache, Chu-Ren Huang, et al. 2020. Comparing Probabilistic, Distributional and Transformer-based Models on Logical Metonymy Interpretation. In *Proceedings of AACL-IJCNLP*.

Giulia Rambelli, Emmanuele Chersoni, Davide Testa, Philippe Blache, and Alessandro Lenci. 2024. Neural Generative Models and the Parallel Architecture of Language: A Critical Review and Outlook. *Topics in Cognitive Science*.

David Samuel, Andrey Kutuzov, Samia Touileb, Erik Velldal, Lilja Øvrelid, Egil Rønningstad, Elina Sigdel, and Anna Palatkina. 2023. NorBench – A Benchmark for Norwegian Language Models. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 618–633, Tórshavn, Faroe Islands. University of Tartu Library.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Alexandra Anna Spalek and Kjell Johan Sæbø. 2019. To Finish in German and Mainland Scandinavian: Telicity and Incrementality. *Journal of Semantics*, 36(2):349–375.

Dan Sperber and Deirdre Wilson. 1986. *Relevance: Communication and Cognition*. Harvard University Press Cambridge, MA.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. ArXiv:2302.13971 [cs].

Matthew J Traxler, Brian McElree, Rihana S Williams, and Martin J Pickering. 2005. Context effects in coercion: Evidence from eye movements. *Journal of Memory and Language*, 53(1):1–25.

Matthew J Traxler, Martin J Pickering, and Brian McElree. 2002. Coercion in sentence processing: Evidence from eye-movements and self-paced reading. *Journal of Memory and Language*, 47(4):530–547.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Zeno Vendler. 1967. *Linguistics in Philosophy*. Cornell University, Ithaca, NY.

Bingyang Ye, Jingxuan Tu, Elisabetta Jezek, and James Pustejovsky. 2022. Interpreting logical metonymy through dense paraphrasing. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.

Alessandra Zarcone and Sebastian Padó. 2011. Generalized event knowledge in logical metonymy resolution. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 33.

Alessandra Zarcone, Sebastian Padó, and Alessandro Lenci. 2014. Logical Metonymy Resolution in a Words-as-Cues Framework: Evidence From Self-Paced Reading and Probe Recognition. *Cognitive Science*, 38(5):973–996.

LLMs Struggle with NLI for Perfect Aspect: A Cross-Linguistic Study in Chinese and Japanese

Jie Lu[1] **Du Jin**[1] **Hitomi Yanaka**[1,2]
[1]the University of Tokyo [2]RIKEN
{lujie2001yoshino, dujin728}@gmail.com
hyanaka@g.ecc.u-tokyo.ac.jp

Abstract

Unlike English, which uses distinct forms (e.g., had, has, will have) to mark the perfect aspect across tenses, Chinese and Japanese lack separate grammatical forms for tense within the perfect aspect, which complicates Natural Language Inference (NLI). Focusing on the perfect aspect in these languages, we construct a linguistically motivated, template-based NLI dataset (1,350 pairs per language). Experiments reveal that even advanced LLMs struggle with temporal inference, particularly in detecting subtle tense and reference-time shifts. These findings highlight model limitations and underscore the need for cross-linguistic evaluation in temporal semantics. Our dataset is available at https://github.com/Lujie2001/CrossNLI.

1 Introduction

Recent advances in large language models (LLMs) have raised important questions about the depth and limits of their language understanding. While these models perform well on many standardized benchmarks, most such evaluations are heavily centered on English and often overlook linguistic features that are specific to other languages.

This paper focuses on whether LLMs have human-like understanding of the perfect aspect of punctual verbs in Chinese and Japanese. Although both languages exhibit features that differ from English (See Section 2.1), there has been no systematic investigation of how the perfect aspect is represented or interpreted in these languages within the NLI framework.

To address this gap, we construct a challenging dataset targeting the interpretation of the perfect aspect with punctual verbs (e.g., *die*) in Chinese and Japanese. Our dataset is linguistically motivated, template-based, and contains 1,350 sentence pairs per language.

Our contributions are as follows:

1. We construct a bilingual NLI dataset focused on perfect aspect in Chinese and Japanese.
2. Our analysis reveals that even the current state-of-the-art LLMs repeatedly fail on specific types of problems in our dataset, indicating that they have not fully acquired a robust or generalizable understanding of the perfect aspect in Chinese and Japanese.

2 Background

2.1 Perfect Aspect in Chinese and Japanese

Following Reichenbach (1947), we analyze the temporal interpretation of the perfect aspect by appealing to a three-way temporal distinction: Speech Time (S), Event Time (E), and Reference Time (R). In Reichenbach's framework, different tenses can be interpreted as different relations between S, E, and R. In the past, R occurs before S; in the present, R and S are simultaneous; in the future, R is after S. Furthermore, in the perfect aspect, E always occurs before R, regardless of tense. In Example (1), E ("Hanako graduates") precedes R ("Taro gets PhD"), thus the overall temporal relation of the sentence is (S $<$ E $<$ R). Here, A $<$ B signifies that A takes place before B.

(1) When Taro gets his PhD next year, Hanako will have graduated from college.

In addition, the time interval between E and R is specified by adding temporal adverbs in the main clause (e.g., "When Taro gets his PhD next year, Hanako will have graduated from college *3 months ago*").

In English, the perfect aspect is marked differently depending on tense (e.g., had, has, will have). However, Chinese and Japanese do not morphologically vary aspect markers across tenses. Chinese typically uses the marker "*-le*(了)"(Klein et al.,

Proceedings of the 16th International Conference on Computational Semantics, pages 99–107
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

2000; Mochizuki, 1997) to indicate the perfect aspect regardless of tense and relies on temporal adverbs or context to convey temporal information. Japanese expresses the perfect aspect using the auxiliary "*-tei-*(-ている-)"(Kudo, 1995; Iori, 2001), combined with either the past "*-tei-ta*(-ていた)" or non-past "*-tei-ru*(-ている)" form, reflecting its binary tense system.[1]

These aspect markers are also used in other contexts and are not exclusively used to express the perfect aspect. For example, Chinese "*le*" may also serve as a modal particle to express urgency or emotional emphasis (e.g., "太好了!" means "great!"). Because such non-perfect uses dominate everyday usage, we hypothesized that LLMs may struggle to generalize the meaning of the perfect aspect in these languages.

2.2 Temporal NLI Datasets

There are already some NLI datasets that focus on aspect (Kober et al., 2019; Pruś et al., 2024). Kober et al. (2019) introduced a carefully curated NLI dataset with a specific focus on tense and aspect. However, these studies focus only on English.

Several studies (Hu et al., 2020; Yanaka and Mineshima, 2021, 2022; Sugimoto et al., 2024) have addressed NLI tasks involving challenging linguistic phenomena in Japanese and Chinese, but they rarely involve NLI tasks focusing on the perfect aspect. OCNLI (Hu et al., 2020) is a Chinese NLI dataset, and JaNLI (Yanaka and Mineshima, 2021) and JSICK (Yanaka and Mineshima, 2022) are Japanese NLI datasets. However, they scarcely address temporal inference. Jamp_sp (Sugimoto et al., 2024) is a Japanese temporal inference dataset, but it does not systematically investigate inference tasks concerning the perfect aspect.

3 Dataset

Based on tense (past (Pst), present (Pres), future (Fut)) and the presence (t) or absence (None) of a temporal adverb in the main clause discussed in Section 2.1, we designed six Japanese sentence templates based on linguistic literature (Kudo, 1995) and created corresponding Chinese templates. By using these sentence templates as premises and hypotheses, we constructed 30 premise–hypothesis pairs (P, H) of NLI problems for Japanese and Chinese, respectively. Since the perfect aspect with punctual verbs expresses a stable temporal relation in sentences, each (P, H) pair is theoretically expected to have a unique correct label (*entailment* or *non-entailment*) under various punctual verb phrases (See a and b in Example (2)). This enables us to generate a large number of (P, H) pairs with entailment labels by inserting different lexical items semi-automatically.

(2) a. Pres(t): Hanako has already **been dead** for 3 months.
⇒ Pres: Hanako has already **been dead**.

b. Pres(t): Hanako has already **graduated from college** for 3 months.
⇒ Pres: Hanako has already **graduated from college**.

The examples of sentence templates with labels for Chinese are shown in Table 1. Full examples of (P, H) pairs (Table 5) and sentence templates in Chinese and Japanese (Tables 6 and Table 7) can be found in Appendix B.

We manually collected 45 sets of common lexical items (nouns and punctual verbs) and clauses to fill our templates. To minimize semantic influence, the items were designed to maintain one-to-one semantic correspondence between Chinese and Japanese. In total, we generated 1,350 (P, H) pairs for each language, comprising 405 instances labeled as entailment and 945 instances labeled as non-entailment.

Some studies have noted that uncertainty may arise in NLI tasks when temporality is involved (Kober et al., 2019; Pavlick and Kwiatkowski, 2019). To address this issue, we limited the verb types to punctual verbs that denote irreversible changes (e.g., *die*).

To validate the reliability of the sentences, all instances in the dataset underwent rigorous review and were refined by native speakers. Additionally, to ensure labeling reliability, multiple native speakers independently annotated 30 different (P, H) pairs. Under a majority voting scheme, their judgments consistently matched the gold labels, demonstrating high inter-annotator agreement.[2]

[1] Other markers such as "guo" (Chinese), "zhe" (Chinese), and "-ta" (Japanese) may express perfect meanings; however, this paper primarily focuses on the prototypical "-le" and "-tei-".

[2] We collected answers from seven native Chinese speakers and three native Japanese speakers. The average match rate between the Chinese responses and the golden label is 94%, while Japanese is 100%.

Categories	Template Example
P: Pst(t)	[Event-Past] 的时候, [NP] 已经 [VP] [TIME] 了.
(E < R < S)	太郎去年取得博士学位 的时候, 花子 已经 死 三个月 了.
	"When Taro got his PhD last year, Hanako *had* already been dead for 3 months."
⇒ H_1: Pst	[Event-Past] 的时候, [NP] 已经 [VP] 了.
(E < R < S)	太郎去年取得博士学位 的时候, 花子 已经 死 了.
	"When Taro got his PhD last year, Hanako *had* already been dead."
⇏ H_2: Pres(t)	[NP] 已经 [VP] [TIME] 了.
(E < S = R)	花子 已经 死 三个月 了.
	"Hanako *has* already been dead for 3 months."
⇒ H_3: Pres	[NP] 已经 [VP] 了.
(E < S = R)	花子 已经 死 了.
	"Hanako *has* already been dead."
⇏ H_4: Fut(t)	[Event-Future] 的时候, [NP] 已经 [VP] [TIME] 了.
(S < E < R)	太郎明年取得博士学位 的时候, 花子 已经 死 三个月 了.
	"When Taro gets his PhD next year, Hanako *will have* already been dead for 3 months."
⇒ H_5: Fut	[Event-Future] 的时候, [NP] 已经 [VP] 了.
(S < E < R)	太郎明年取得博士学位 的时候, 花子 已经 死 了.
	"When Taro gets his PhD next year, Hanako *will have* already been dead."

Table 1: Template examples of premise and hypothesis sentences in Chinese. In category column, the symbol (t) indicates the presence of a temporal adverb in the main clause. The slot [Event-Past] and [Event-Future] is a subordinate clause containing a temporal expression referring to the past or future, such as "太郎去年取得博士学位" ("Taro got his PhD last year"). ⇒ indicates *entailment* and ⇏ indicates *non-entailment*.

4 Experimental Setup

We conducted experiments on multilingual LLMs and LLMs with enhanced monolingual capability with varying parameter scales. The multilingual models we used include GPT-4 (gpt-4-0613), Claude 3.5 (claude-3-5-sonnet-20241022), Deepseek-V3 (deepseek-chat), and Llama3.1[3] (8B and 70B). The LLMs with enhanced monolingual capability include the Chinese models Qwen3[4] (8B and 32B) and the Japanese models Swallow[5] (9B and 27B). These models cover both multilingual and language-specialized types.

Each model received every premise–hypothesis pair in the corresponding language, together with an instructional prompt that introduces the NLI task and asks whether the premise entails the hypothesis. Model predictions were then compared with gold labels to compute classification accuracy. All experiments were conducted in a zero-shot setting. Our Japanese prompts were adapted from (Sugimoto et al., 2024) and then translated into Chinese by native speakers. The full Chinese and Japanese prompts are provided in Appendix A.

[3] hf.co/collections/meta-llama/llama-31-669fc079a0c406a149a5738f

[4] hf.co/collections/Qwen/qwen3-67dd247413f0e2e4f653967f

[5] hf.co/collections/tokyotech-llm/gemma-2-swallow-67f2bdf95f03b9e278264241

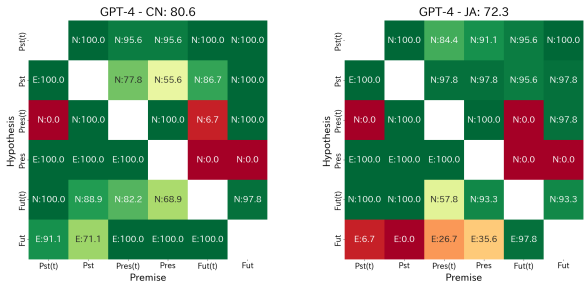


Figure 1: Detailed results from GPT-4 in Chinese and Japanese.The overall accuracy is shown in the title. E/N:number in cells shows the gold label and the accuracy for each (P, H) pair.

5 Results and Discussion

Table 4 shows the average accuracy of tested models on our dataset. Figure 1 shows the detailed results of GPT-4. See Appendix C for detailed results of other models.

Comparison between models As shown in Table 4, Claude 3.5 achieved the best overall performance, outperforming GPT-4—the second-best model—by over 10% in both Chinese and Japanese.

Most models performed similarly on Chinese and Japanese, with accuracy differing by less than 5%. However, Llama-8B was a notable outlier, showing a large performance gap of 26.2% (Chinese: 37.3%, Japanese: 65.6%). Notably, Llama-

Tense of (P, H)	Label	GPT-4	Claude3.5	Deepseek-v3	Llama-8B	Llama-70B	Qwen3-8B	Qwen3-32B	Swallow-9B	Swallow-27B
(Pst(t), Pres(t))	**N**	**0.0/0.0**	**77.8/2.2**	**0.0/0.0**	**0.0/17.8**	**0.0/0.0**	**0.0/0.0**	**0.0/0.0**	**0.0/0.0**	**0.0/0.0**
(Pst(t), Pres)	**E**	100.0/100.0	100.0/97.8	100.0/100.0	100.0/57.8	100.0/95.6	100.0/100.0	100.0/100.0	95.6/62.2	100.0/62.2
(Pst, Pres(t))	N	100.0/100.0	100.0/100.0	100.0/100.0	0.0/100.0	93.3/80.0	91.1/62.2	51.1/75.6	93.3/62.2	15.6/62.2
(Pst, Pres)	**E**	**100.0/100.0**	**95.6/84.4**	**100.0/100.0**	**100.0/62.2**	**100.0/88.9**	**100.0/100.0**	**100.0/100.0**	**100.0/60.0**	**100.0/60.0**
(Fut(t), Pres(t))	**N**	**6.7/0.0**	**91.1/24.4**	**0.0/0.0**	**0.0/51.1**	**8.9/8.9**	**0.0/0.0**	**0.0/2.2**	**0.0/0.0**	**0.0/0.0**
(Fut(t), Pres)	**N**	**0.0/0.0**	**53.3/20.0**	**0.0/0.0**	**2.2/62.2**	**0.0/37.8**	**0.0/0.0**	**2.2/2.2**	**13.3/4.4**	**6.7/8.9**
(Fut, Pres(t))	N	100.0/97.8	100.0/100.0	100.0/100.0	11.1/95.6	97.8/93.3	97.8/46.7	48.9/82.2	97.8/62.2	51.1/60.0
(Fut, Pres)	**N**	**0.0/0.0**	**86.7/42.2**	**2.2/0.0**	**0.0/51.1**	**0.0/73.3**	**0.0/0.0**	**0.0/33.3**	**0.0/24.4**	**2.2/15.6**

Table 2: Model accuracy (%) when the premise is in the past or future, and the hypothesis is in the present tense. Left side of "/" shows accuracy in Chinese cases, and the right side shows Japanese cases. E indicates entailment labels and N indicates non-entailment labels. The rows in boldface indicate the questions with lexical overlap.

Model	Language	Accuracy (E / N)
Llama-8B	CN	**92.6% / 13.5%**
	JA	45.2% / 71.3%
Qwen3-8B	CN	44.6% / 74.6%
	JA	62.7% / 71.4%
Swallow-9B	CN	46.9% / 80.2%
	JA	32.6% / 48.4%

Table 3: The differences in accuracy between *entailment* and *non-entailment* cases for Llama-8B, Qwen3-8B and Swallow-9B.

Model	Accuracy (CN / JA)
GPT-4	80.6% / 72.3%
Claude3.5	91.5% / 76.7%
Deepseek-v3	77.3% / 70.1%
Llama-8B	37.3% / 65.6%
Llama-70B	75.8% / 72.3%
Qwen3-8B	74.2% / 68.8%
Qwen3-32B	51.4% / 56.6%
Swallow-9B	70.2% / 43.6%
Swallow-27B	54.9% / 42.7%

Table 4: Overall accuracy of each model on our dataset.

8B shows an accuracy gap of nearly 80% between instances labeled as entailment and those labeled as non-entailment (See Table 3). Given that the contexts in which the perfect aspect appears in Chinese are more homogeneous, this result suggests that multilingual models with smaller parameter sizes may struggle to generalize the meaning of the perfect aspect in Chinese.

Furthermore, LLMs with enhanced monolingual capability (Qwen3 and Swallow) exhibit a negative correlation between accuracy and model size. We aim to explore this phenomenon in greater depth in future studies.

Comparison based on linguistic phenomena When the tense of the premise and the hypothesis is the same, models with parameter sizes over 32 billion achieve near-perfect accuracy, while those with lower parameter sizes still struggle with it. Example (3) shows a case of (P: Pst(t), H: Pst).

(3) Pst(t): 太郎上周回到家的时候，花子已经死三天了。
"When Taro came home last week, Hanako had already been dead for 3 days."
$\not\Rightarrow$ Pst: 太郎上周回到家的时候，花子已经死了。
"When Taro came home last week, Hanako had already been dead."

This demonstrates that models with larger parameter sizes can capture the semantic nuances introduced by temporal adverbs.

However, when the tense of the premise and the hypothesis differ, the situation becomes more complex. In cases where the premise is the past or future and the hypothesis is the present (e.g., (P: Fut, H: Pres), we found all models except Claude3.5 consistently predict *entailment* (See Table 2). One possible reason is that the models rely on lexical overlap heuristics mentioned in (McCoy et al., 2019) to solve these problems. In Chinese, since the aspect marker "*le*" applies across all tenses, lexical overlap naturally occurs. In Japanese, sentence pairs where both the premise and the hypothesis use the same perfect aspect marker (e.g., (Fut, Pres)) involve lexical overlap. Examples (4) and (5) illustrate cases where lexical overlap occurs.

(4) Fut: 太郎明年大学毕业的时候，花子已经辞职了。
"When Taro graduates from college next year, Hanako will have already quit her job."
$\not\Rightarrow$ Pres: 花子已经辞职了。
"Hanako has already quit her job."

(5) Fut: 太郎が来年大学を卒業するとき、花子はとっくに会社を辞めている。
"When Taro graduates from college next year, Hanako will already have quit her job."
$\not\Rightarrow$ Pres: 花子は会社を辞めている。
"Hanako has already quit her job."

To our surprise, in Japanese cases where the premise and hypothesis use different tense markers, models still tend to incorrectly predict *entailment*, as illustrated by Example (6), in which "*-tei-ta*" is used in the premise and "*-tei-ru*" in the hypothesis. This result may suggest that the models' low accuracy in handling the perfect aspect in both Chinese and Japanese is not merely a consequence of heuristic biases, but also reflects their incomplete understanding of the semantic distinction between the Japanese perfect aspect marker "*-tei-ta*" and the simple past marker "*-tei-ru*".

(6) Pst(t): 太郎が先週に家に帰ったとき、花子は既に三日前に死んでいた。
"When Taro came home last week, Hanako had already been dead for 3 days."
$\not\Rightarrow$ Pres(t): 花子は三日前に死んでいる。
"Hanako has already been dead for 3 days."

6 Conclusion

In this study, we presented a bilingual NLI dataset targeting the interpretation of the perfect aspect with punctual verbs in Japanese and Chinese. Our results show that even state-of-the-art LLMs often fail to capture the correct temporal relations, especially when tense and reference times differ between sentences. Our findings highlight the need for evaluation benchmarks that are both linguistically diverse and sensitive to temporal inference.

7 Limitation and Future Work

One limitation of this study is that our experiments deliberately include only punctual, irreversible verbs (e.g., die) to avoid truth-conditional ambiguities. Consequently, our findings do not yet generalize to verbs that occur in perfect-progressive constructions. Extending coverage to such verb classes is left for future work.

Another limitation is that our experiments are only performed in a zero-shot setting. We plan to expand the range of prompt formats used in future experiments.

Finally, some phenomena highlighted in Section 5 remain speculative, most notably the negative scaling trend observed for the Qwen3 series and the Swallow series. We will design additional controlled experiments to validate or refute these hypotheses.

8 Acknowledgement

We would like to express our heartfelt gratitude to Koki Shibata, Tomoki Doi, Taiga Someya, Daiki Matsuoka, and Izumi Konishi for their generous support and invaluable assistance in both the discussions and the writing process of this research. Without them, this work would not have been possible. We also thank the three anonymous reviewers for their helpful comments and feedback. This work was partially supported by the Institute for AI and Beyond of the University of Tokyo, and JSPS KAKENHI grant number JP24H00809.

References

Hai Hu, Kyle Richardson, Liang Xu, Lu Li, Sandra Kuebler, and Larry Moss. 2020. Ocnli: Original chinese natural language inference. In *Findings of EMNLP*.

Isao Iori. 2001. Teirukei, teitakei no imi no toraekata ni kansuru hitoshian (a proposal for the proper understanding of the meaning of the aspectual morpheme -tei in japanese). *Hitotsubashi Daigaku Ryuugakusei Sentaa Kiyoo (Bulletin of the Center for International Education, Hitotsubashi University)*, 4:75–94.

W. Klein, Ping Li, and H. Hendriks. 2000. Aspect and assertion in mandarin chinese. *Natural Language and Linguistic Theory*, 18(4):723–770.

Thomas Kober, Sander Bijl de Vroe, and Mark Steedman. 2019. Temporal and aspectual entailment. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 103–119, Gothenburg, Sweden. Association for Computational Linguistics.

Mayumi Kudo. 1995. *Aspect and Tense System and Text: Temporal Expressions in Modern Japanese*. Japanese Language Research Series, 2nd Series, Vol. 7. Hitsuji Shobo. Originally in Japanese.

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.

Keiko Mochizuki. 1997. Chuugokugo no paafekuto sou (the perfect aspect in chinese). *Toukyou Gaikokugo Daigaku Ronshuu (Tokyo University of Foreign Studies Journal)*, 55:55–71.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Katarzyna Pruś, Mark Steedman, and Adam Lopez. 2024. Human temporal inferences go beyond aspectual class. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1913–1923, St. Julian' s, Malta. Association for Computational Linguistics.

Hans Reichenbach. 1947. *Elements of Symbolic Logic*. Macmillan.

Tomoki Sugimoto, Yasumasa Onoe, and Hitomi Yanaka. 2024. Jamp_sp : A controlled japanese temporal inference dataset considering aspect. *Journal of Natural Language Processing*, 31(2):637–679.

Hitomi Yanaka and Koji Mineshima. 2021. Assessing the generalization capacity of pre-trained language models through Japanese adversarial natural language inference. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 337–349, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hitomi Yanaka and Koji Mineshima. 2022. Compositional evaluation on Japanese textual entailment and similarity. *Transactions of the Association for Computational Linguistics*, 10:1266–1284.

A Prompts

Chinese:
指示: 从 entailment, non-entailment 中回答前提和假设的关系.不需要给出解释.
限制：
- 如果能够通过逻辑知识或常识性知识从前提推导出假设，则输出 entailment.
- 如果前提成立无法保证假设成立,则输出 non-entailment.
- 前提和假设中没有省略任何时间成分.
- 前提和假设的发话时点为现在.
前提: {premise}
假设: {hypothesis}
答案:

Japanese:
指示: 前提と仮説の関係を entailment,non-entailment の中から回答してください.説明は不要です.
制約：
- 前提から仮説が,論理的知識や常識的知識を用いて導出可能である場合は entailment と出力
- 前提が成り立つとしても仮説が必ずしも成り立たない場合は non-entailment と出力
- 前提と仮説には,時間的な成分を省略していない
- 前提と仮説の発話時を現在とする
前提: {premise}
仮説: {hypothesis}
答え:

English translation:
Instruction: Answer the relationship between the premise and the hypothesis with one of the following: entailment or non-entailment. No explanation is needed.
Constraints:
- If the hypothesis can be deduced from the premise through logical reasoning or common sense knowledge, output entailment.
- If the truth of the premise does not guarantee the truth of the hypothesis, output non-entailment. - There is no omission of any temporal information in both the premise and hypothesis.
- The utterance time for both the premise and hypothesis is the present.
Premise: {premise}
Hypothesis: {hypothesis}
Answer:

B Templates

Table 5 shows all (P, H) templates and their labels in our dataset. Table 6 and Table 7 show Chinese and Japanese sentence templates used to create our dataset.

C Detailed Results

Figure 2 and Figure 3 show detailed results of all models under our dataset.

Premise	Hypothesis	Example	Label
Pst(t)		When Taro got his PhD last year, Hanako had already been dead for 3 months.	
	Pst	When Taro got his PhD last year, Hanako had already been dead.	Entailment
	Pres(t)	Hanako has already been dead for 3 months.	Non-Entailment
	Pres	Hanako has already been dead.	Entailment
	Fut(t)	When Taro gets his PhD next year, Hanako will have already been dead for 3 months.	Non-Entailment
	Fut	When Taro gets his PhD next year, Hanako will have already been dead.	Entailment
Pst		When Taro got his PhD last year, Hanako had already been dead.	
	Pst(t)	When Taro got his PhD last year, Hanako had already been dead for 3 months.	Non-Entailment
	Pres(t)	Hanako has already been dead for 3 months.	Non-Entailment
	Pres	Hanako has already been dead.	Entailment
	Fut(t)	When Taro gets his PhD next year, Hanako will have already been dead for 3 months.	Non-Entailment
	Fut	When Taro gets his PhD next year, Hanako will have already been dead.	Entailment
Pres(t)		Hanako has already been dead for 3 months.	
	Pst(t)	When Taro got his PhD last year, Hanako had already been dead for 3 months.	Non-Entailment
	Pst	When Taro got his PhD last year, Hanako had already been dead.	Non-Entailment
	Pres	Hanako has already been dead.	Entailment
	Fut(t)	When Taro gets his PhD next year, Hanako will have already been dead for 3 months.	Non-Entailment
	Fut	When Taro gets his PhD next year, Hanako will have already been dead.	Entailment
Pres		Hanako has already been dead.	
	Pst(t)	When Taro got his PhD last year, Hanako had already been dead for 3 months.	Non-Entailment
	Pst	When Taro got his PhD last year, Hanako had already been dead.	Non-Entailment
	Pres(t)	Hanako has already been dead for 3 months.	Non-Entailment
	Fut(t)	When Taro gets his PhD next year, Hanako will have already been dead for 3 months.	Non-Entailment
	Fut	When Taro gets his PhD next year, Hanako will have already been dead.	Entailment
Fut(t)		When Taro gets his PhD next year, Hanako will have already been dead for 3 months.	
	Pst(t)	When Taro got his PhD last year, Hanako had already been dead for 3 months.	Non-Entailment
	Pst	When Taro got his PhD last year, Hanako had already been dead.	Non-Entailment
	Pres(t)	Hanako has already been dead for 3 months.	Non-Entailment
	Pres	Hanako has already been dead.	Non-Entailment
	Fut	When Taro gets his PhD next year, Hanako will have already been dead.	Entailment
Fut		When Taro gets his PhD next year, Hanako will have already been dead.	
	Pst(t)	When Taro got his PhD last year, Hanako had already been dead for 3 months.	Non-Entailment
	Pst	When Taro got his PhD last year, Hanako had already been dead.	Non-Entailment
	Pres(t)	Hanako has already been dead for 3 months.	Non-Entailment
	Pres	Hanako has already been dead.	Non-Entailment
	Fut(t)	When Taro gets his PhD next year, Hanako will have already been dead for 3 months.	Non-Entailment

Table 5: All (P, H) templates and their labels. Here, we only present the English translation of one example to illustrate the correspondence between the (P, H) pair and their label in our dataset. As mentioned in Section 3, the label remains unchanged even when different punctual verbs are used.

Category	Template	Example
Pst(t)	[Event-Past]的时候，[NP]已经[VP][TIME]了	田中上周搬家的时候，山本已经合格大学一周了
Pst	[Event-Past]的时候，[NP]已经[VP]了	田中上周搬家的时候，山本已经合格大学了
Pres(t)	[NP]已经[VP][TIME]了	山本已经合格大学一周了
Pres	[NP]已经[VP]了	山本已经合格大学了
Fut(t)	[Event-Future]的时候，[NP]已经[VP][TIME]了	佐藤下个月工作的时候，山本已经合格大学一周了
Fut	[Event-Future]的时候，[NP]已经[VP]了	佐藤下个月工作的时候，山本已经合格大学了

Table 6: Sentence Templates for Chinese.

Category	Template	Example
Pst(t)	[Event-Past]とき、[NP]は[TIME]前にすでに[V-teita]	田中が先月引っ越したとき、山本は一週間前にすでに大学に合格していた
Pst	[Event-Past]とき、[NP]はすでに[V-teita]	田中が先月引っ越したとき、山本はすでに大学に合格していた
Pres(t)	[NP]は[TIME]前に[V-teiru]	山本は一週間前に大学に合格している
Pres	[NP]は[V-teiru]	山本は大学に合格している
Fut(t)	[Event-Future]とき、[NP]は[TIME]前に[V-teiru]	佐藤が来月転職するとき、山本は一週間前に大学に合格している
Fut	[Event-Future]とき、[NP]はとっくに[V-teiru]	佐藤が来月転職するとき、山本はとっくに大学に合格している

Table 7: Sentence Templates for Japanese.

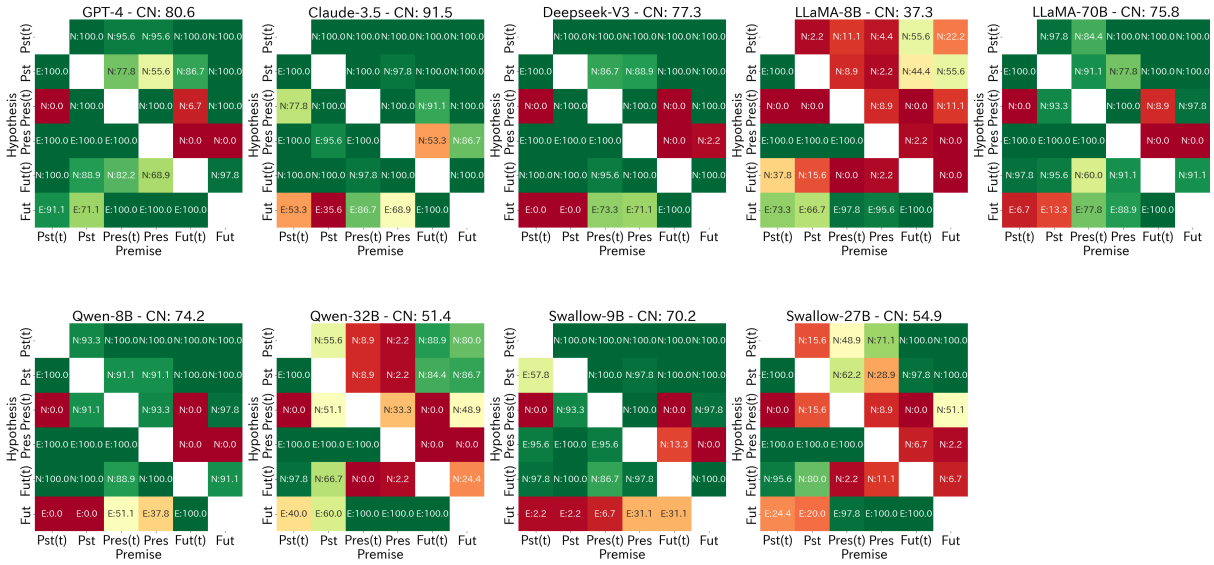


Figure 2: Results on our Chinese dataset. The overall accuracy is shown in the title. E/N:number in cells shows the gold label and the accuracy for each (P, H) pair.

Figure 3: Results on our Japanese dataset. The overall accuracy is shown in the title. E/N:number in cells shows the gold label and the accuracy for each (P, H) pair.

Assessing LLMs' Understanding of Structural Contrasts in the Lexicon

Shuxu Li
OLST, Université de Montréal
shuxu.li@umontreal.ca

Antoine Venant
OLST, Université de Montréal
antoine.venant@umontreal.ca

Philippe Langlais
RALI, Université de Montréal
felipe@iro.umontreal.ca

François Lareau
OLST, Université de Montréal
francois.lareau@umontreal.ca

Abstract

We present a new benchmark to evaluate the lexical competence of large language models (LLMs), built on a hierarchical classification of lexical functions (LFs) within the Meaning-Text Theory (MTT) framework. Based on a dataset called *French Lexical Network* (LN-fr), the benchmark employs contrastive tasks to probe the models' sensitivity to fine-grained paradigmatic and syntagmatic distinctions. Our results show that performance varies significantly across different LFs and systematically declines with increased distinction granularity, highlighting current LLMs' limitations in relational and structured lexical understanding.

1 Introduction

Large language models (LLMs) like GPT-4 (OpenAI et al., 2024), Qwen (Bai et al., 2023), or LLaMA (Touvron et al., 2023) do not merely generate coherent text. They can be prompted to solve a wide range of linguistic and cognitive tasks, such as question answering, information extraction, or machine translation, with remarkable performance (Zhao et al., 2025). As a result, works on LLMs' evaluation have shifted focus away from grammaticality and coherence, towards reasoning capacities, factual consistency, bias, or other **extra-linguistic properties** (Chang et al., 2023).

Yet, there remain essential questions about the nature and depth of linguistic knowledge captured by these models and their ability to introspectively access and share this knowledge. While LLMs appear to "use language" fluently, the amount of linguistic structure they "understand" is not clearly circumscribed, nor is their ability to reason abstractly about linguistic objects.

The **lexicon** is a case in point. A proper understanding of language necessarily entails a grasp of its lexicon—not as a mere inventory of words and their definitions, but as a structured system wherein lexical units are interconnected through a variety of **relations** (like synonymy, antonymy, morphological derivations, intensification, and others) that recur across most (if not all) languages. Leveraging such relations to assess linguistic competence has long been an attractive idea: they are, for instance, at the heart of popular analogical benchmarks (Turney et al., 2004; Mikolov et al., 2013; Gladkova et al., 2016, *inter alia*) which have become a staple of the evaluation of distributional representations. However, these analogical datasets arguably lack both theoretical grounding and coverage in some areas. For instance, the Bigger Analogy Test Set (Gladkova et al., 2016), one of the most balanced, diverse and challenging benchmarks, covers very few *syntagmatic* (*i.e.* related to word *combinations* rather than word *substitutions*) lexical relations and leaves out many aspects related to meaning rather than strict morphology (like the analogy between the pairs *continue:continuation::sell:sale*).

We therefore wish to ground an evaluation benchmark on a well-established lexicographic theory: the **Meaning-Text Theory (MTT)** (Mel'čuk, 1973, 1996, 2016; Mel'čuk and Polguère, 2021). MTT places the lexicon and its combinatorial properties at the core of linguistic modeling. To formally model the structure of the lexicon, MTT uses a system of *Lexical Functions* (LFs), which represent consistent and recurrent paradigmatic or syntagmatic relations between *lexical units*—that is, words taken in a specific sense. Each LF encodes a specific semantic or syntactic relation between a lexical unit (its *keyword*) and a set of lexical units (its *value*). The following examples illustrate some of the most common LFs[1] :

[1] In line with MTT's notational conventions, we overload the = symbol to denote set membership rather than equality. Thus $f(a) = b$ means in fact $b \in f(a)$, as an LF typically associates a keyword with more than one value. One has for instance Syn(*film*) = *movie* and Syn(*film*) = *picture*.

Proceedings of the 16th International Conference on Computational Semantics, pages 108–119
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

- Syn(*film*) = *movie* (synonym)
- Magn(*awake*) = *wide* [~] (intensifier)
- Oper_2(*criticism*) = *(to) face* [~] (support verb)[2]

The question we ask is how accurately LLMs can be prompted to recognize whether a pair of French words instantiates a given type of lexical relation. To answer this question, we build on MTT and define a set of target LFs of interest, capturing lexical knowledge **at different levels of granularity**. For instance, at a coarse level, we test whether the LLM can tell apart instances of adjectival derivations (of any kind) from instances of other type of derivations (*e.g.* nominal, or verbal ones), and at a finer level, whether it can discriminate rather semantically neutral adjectival derivations (like *destroy–destructive*) from those involving a stronger meaning shift (like *destroy–destructible*). To this aim, we associate each target LF with a set of *contrastive* LFs, so that each contrastive LF both share a common property with the target (*e.g.* both correspond to some kind of adjectival derivation) and are distinguished by another property (*e.g.* they correspond to different degrees or types of meaning shifts), and ask LLMs to recognize the pairs of words obtained from the target and reject those obtained from its contrastive LFs. To automatically obtain the pairs of words, we leverage a high quality French lexicographic resource, the *French Lexical Network* (Lux-Pogodalla and Polguère, 2011; ATILF, 2024, henceforth, LN-fr), which offers extensive coverage and is closely aligned with the theoretical framework adopted here. Although we use French data, the lexical relations we target are universal. We work from the assumption that if a model performs well on French, it should perform about as well on other languages similarly covered by its pretraining material.

We thus contribute a *hierarchy* of LFs, wherein each intermediate level corresponds to some coarse-grained lexical relation (such as 'verbal collocation'), and immediate descendants correspond to distinct sub-relations of the former (such as 'support verbs' and 'semantically loaded verbal collocations'). We propose a benchmark of polar questions to test LLMs' ability to specifically recognize these contrasts, and assess several open-weights LLMs on this benchmark, as well as the effect of different prompting configurations. We also investigate the impact of surface cues on the LLM's behavior.

[2]Support verbs serve to build a syntactically well-formed structure without contributing additional meaning (Mel'čuk and Polguère, 2021; Ramos and Tutin, 1996).

2 Related work

The semantic abilities of computational models have often been measured by their ability to recognize or perform analogies. Analogical datasets such as SAT (Turney et al., 2004), the Google analogy test set (Mikolov et al., 2013), and BATS (Gladkova et al., 2016) have become popular benchmark of this capacity. They also have been applied to the evaluation of recent LLMs' semantic abilities: Ushio et al. (2021) evaluate LLMs on well-established analogical benchmarks using prompts and their completion probabilities, and show, among many other things, that the lexical analogies of BATS are more difficult for the models than the morphological or encyclopedic ones. Yuan et al. (2024) show that automatically extracting analogies from a knowledge graph can be used to enhance LLMs performance on analogical benchmarks *via* fine-tuning or few-shot learning.

Some new benchmarks have also been developed: Wijesiriwardene et al. (2023) introduce a benchmark of analogies between longer texts, targeting concepts such as entailment or explanation, and Chen et al. (2022) introduce a benchmark of exam problems and associated analogical reasoning. While these resources are important tools to assess higher level linguistic and reasoning capabilities, they also steer away from evaluating the sheer *lexical* competence of language models. Other approaches have taken inspiration from psycholinguistic methods like cloze completion tasks. Some of the tasks considered in (Ettinger, 2020) directly concerns lexical knowledge. They find that BERT (Devlin et al., 2019) is better at recognizing hypernyms than distinguishing semantic roles.

While models' mastery of *paradigmatic* relations such as synonymy or hyponymy is extensively tested in the aforementioned works, the type of knowledge underlying support or light verb constructions (like *chance* and *take*), or tied to the argument structure (*doctor* and *patient*) is more often overlooked. Our work addresses this gap with a benchmark exclusively centered around the lexicon, allowing a *systematic* and *granular* exploration of LLMs' ability to recognize the whole range of lexical functions formally defined by Meaning-Text linguists. It is akin to the recent work of Petrov et al. (2025), who have also leveraged instances of LFs from LN-fr to diagnose lexical competence,

but supplements theirs in several respects. Petrov et al. (2025) designed a challenging analogy-based benchmark of 2,600 fine-grained lexical analogies using 25 common LFs (21 paradigmatic and 4 syntagmatic), and showed that moderately-sized LLMs achieve particularly strong performance on derivational morphology but struggle more with syntagmatic relations and distinguishing event-participant roles. In contrast, we organize relations in a system of hierarchical clusters, grouping specific relations into broader categories, and examine models' ability to make distinctions with variable levels of specificity. Rather than directly requesting models to solve a given analogical equation (an open question), we use closed yes/no questions with more elaborate contexts. While this arguably makes the task less challenging, it also circumvents important shortcomings of bare analogical equations regarding the amount of information provided to LLMs, and makes it easier to avoid false negatives in the evaluation. In particular, it enables us to include information pertaining to word sense clarification and/or semantic roles indices in the prompts, and thereby study a wider and more balanced range of lexical relations.

3 Evaluation Framework

This section outlines the evaluation framework designed for our study, including our proposal of a hierarchical organization of LFs, the lexical dataset, and the construction of contrastive prompts.

3.1 Hierarchical structure of LFs

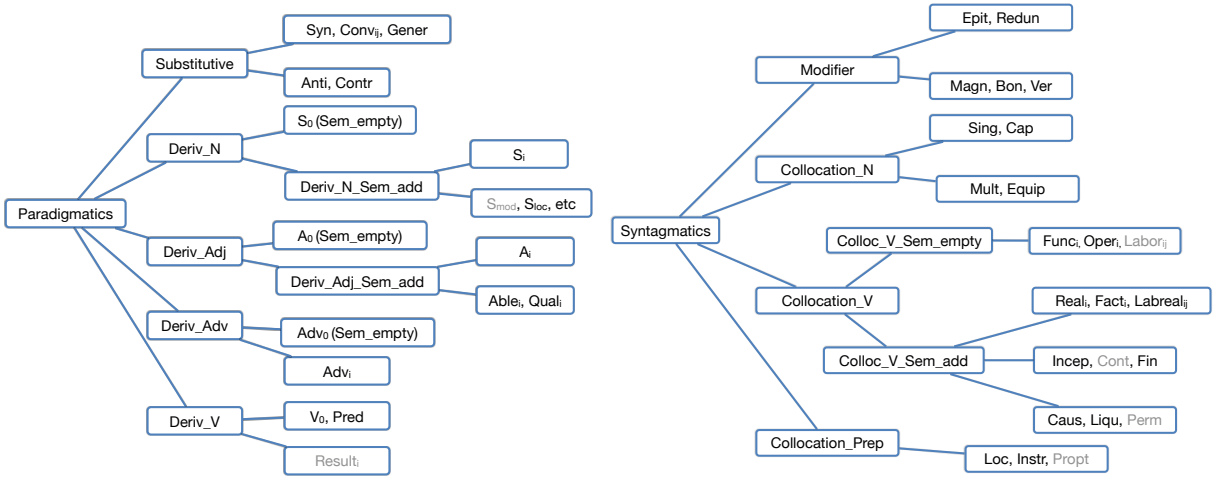
In the MTT framework and its associated resources, the instances of LFs are highly specific. For example, $\mathtt{S}_0(produce_{\mathrm{V}})$ refers to the abstract activity denoted by the verb itself, yielding the nominal form *production*, and thus represents a derivation without added semantic content. In contrast, $\mathtt{S}_1(produce_{\mathrm{V}})$ yields *producer*, designating the agent of the activity—the first argument of the predicate 'produce$_{\mathrm{V}}$'. Similarly, $\mathtt{S}_2(produce_{\mathrm{V}})$ yields *product*, referring to the result of the activity—the second argument of the predicate.

This illustrates two levels of semantic distinction: while $\mathtt{S}_1$ and $\mathtt{S}_2$ are both argument-oriented derivations and thus semantically close, they differ based on which argument role they instantiate. $\mathtt{S}_0$, on the other hand, is more distinct as its value encodes the event itself without any further semantic shift. In the present study, we are particularly interested in whether LLMs are sensitive to distinctions among LFs at varying levels of granularity. To systematically assess their lexical competence in this regard, a structured classification scheme is required for explicitly modeling such fine-grained distinctions.

Building on the theoretical foundations of LFs in MTT (Mel'čuk, 1996; Ramos and Tutin, 1996; Jousse, 2010; Mel'čuk and Polguère, 2021), we first classify the full set of *Simple Standard LFs* according to their semantic and syntactic properties. At the top level, we distinguish between paradigmatic LFs (encoding derivational or synonymic relations) and syntagmatic LFs (encoding collocation patterns). Each group is further subdivided by the part of speech (POS) of the keyword and the value. Within these groups, finer-grained categories are defined according to specific semantic properties. In particular, certain distinctions between LFs arise from subtle syntactic differences in the realization of the semantic arguments associated with the keyword. These cases are categorized more finely. For example, within the category of *Nominal Derivation*, $\mathtt{S}_0$ denotes purely syntactic derivation without any semantic enrichment, whereas $\mathtt{S}_i$ represents the noun that refers to typical semantic arguments of the keyword. The $\mathtt{S}_i$ category itself can be further subdivided. In particular, $\mathtt{S}_1$ returns the name of the first semantic argument of the keyword, e.g., $\mathtt{S}_1$(*sell*) = *seller*, while $\mathtt{S}_2$ corresponds to the second, e.g., $\mathtt{S}_2$(*sell*) = *merchandise*. This hierarchical classification of LFs, as illustrated in Figure 1, is structured at multiple levels of granularity and serves as the foundation for our evaluation of lexical competence in LLMs.

3.2 Data

The MTT framework has given rise to a substantial body of lexicographic work, including Mel'čuk et al. (1995); Apresjan (2000); Mel'čuk et al. (1999); Mangeot (2000); Polguère (2014); Alonso Ramos (2015); L'Homme et al. (2009); Barrios Rodríguez (2024). Among them, the *French Lexical Network* (LN-fr) (Lux-Pogodalla and Polguère, 2011; ATILF, 2024) stands out as a large-scale lexical network where nodes represent French lexical units and edges encode syntagmatic or paradigmatic LFs, as Figure 2 demonstrates. In the present study, prompt generation for model evaluation relies on the lexicographic resource LN-fr (Lux-Pogodalla and Polguère, 2011; ATILF, 2024).



(a) Paradigmatic LFs

(b) Syntagmatic LFs

Figure 1: Hierarchical classification of *Simple Standard LFs*. LFs shown in grey are theoretically part of the hierarchy but are excluded from the evaluation due to insufficient instances in the dataset. For details on definitions of terminal-node LFs, see (Mel'čuk and Polguère, 2021)

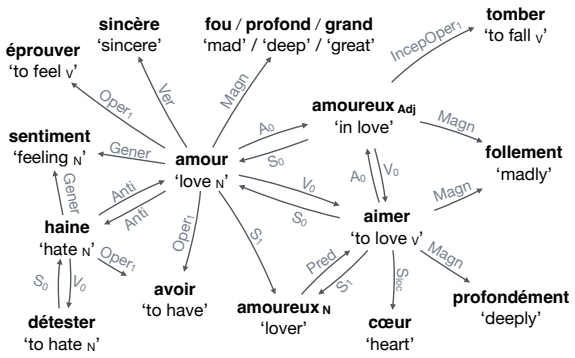


Figure 2: LN-fr example for lexical unit *amour* 'love' and its relations with other lexical units.

Built according to the methodological principles of Explanatory Combinatorial Lexicology (Mel'čuk et al., 1995), it comprises ~30k lexical units covering ~19k lemmas in French. In addition to propositional forms and usage examples, LN-fr includes over 66k annotated instances of LFs, forming a rich network of paradigmatic and syntagmatic relations.

A node in our hierarchical structure corresponds to a group of LF instances drawn from the LN-fr dataset. We retained only instances with complete information, including the LF identifier, the *keyword* (input lexical unit), and the *value* (output lexical unit). Any instance missing one of these fields was excluded. The resulting filtered dataset served as the sampling pool for prompt construction during evaluation. To ensure sufficient coverage and statistical reliability, we further **excluded** all LF nodes with **fewer than 30** valid instances from the final evaluation set, which are represented in grey in Figure 1. The full hierarchical structure, including both terminal and intermediate nodes, is specified in a dedicated configuration file, following the theoretical principles outlined in Mel'čuk et al. (1995); Mel'čuk and Polguère (2021).

3.3 Contrastive Sampling and Prompting

Building on the *Natural Instructions* paradigm, which enables model interaction through prompt-based question answering enriched with few-shot demonstrations and contrastive examples (Mishra et al., 2022; Chang et al., 2023), we adopt a contrastive sampling strategy to evaluate LLMs' ability to distinguish lexical relations. Grounded in our hierarchical classification of LFs, each prompt presents a balanced set of positive and negative examples centered on a target LF category.

To generate negative examples, we sample contrasting instances from sibling nodes under the same parent within the LF hierarchy, ensuring functional but structurally proximate distinctions, as shown in Figure 3. For example, if the node `Substitutive` in our hierarchy (see Figure 1) is selected as the target, all its sibling nodes (e.g. `Deriv_N`, `Deriv_Adj`) are considered contrasts.

Prompt Our evaluation strategy follows the paradigm of Prompt Engineering (Schulhoff et al.,

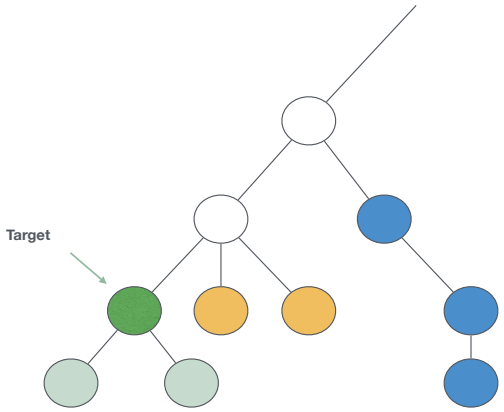


Figure 3: Contrastive sampling: positives from the target LF (green); negatives from yellow nodes as contrasts.

2025), in particular the *Natural Instructions* framework (Mishra et al., 2022), where models are prompted with structured input-output examples in natural language. For each target LF, we construct multiple prompts, each of which encodes a distinct contrastive setup based on instance sampling.

```
As a linguist expert in the Meaning-Text Theory,
you will be given a definition of a lexical
function, along with a set of positive and negative
examples. Then, you will be presented with a new
pair of keyword and value, and your task is to
answer 'Yes' if the pair corresponds to the target
LF, or 'No' if it does not [...]
```

Listing 1: System Prompt

As illustrated in Listings 1 and 2, the *System prompt* provides the overall task description and specifies the expected output format. The *User prompt*, in turn, introduces the target LF through a formal definition, followed by a set of positive and negative examples. For each example, we present the surface forms of the keyword and its value, along with the propositional form of the keyword. Optionally, the prompt also includes a KWIC (keyword in context) snippet for the keyword—a 13-word window centered on the keyword—and the propositional form of the value. The propositional form is a minimal example phrase involving the keyword and numbered placeholders, whose purpose is to describe the conventional numbering of semantic arguments and their correspondence with syntactic positions in an example. For instance, the propositional form for *sale* could be *~ carried out by $1 to $2 for the amount $3* (where ~ links to the keyword, *sale*). This propositional form would

```
Oper_1 is a lexical function which, given a lexical
unit as a keyword, selects another one as a
collocate in order to form a lexical collocation...

Here are some positive examples of this function:
fatigue -> éprouver
Propositional form of the keyword: ~de $1 causé par
$2
KWIC context of the keyword: ...
Answer: Yes
...

Here are some negative examples of this function:
cheveu -> soigner
Propositional form of the keyword: ~de $1
KWIC context of the keyword: ...
Answer: No
...

QUESTION:
football -> jouer
Propositional form of the keyword: ~pratiqué par $1
KWIC context of the keyword: ...

Does the above word pair also constitute a valid
example of this class of lexical function?
```

Listing 2: User Prompt

indicate that the seller is conventionally considered the first semantic argument, the buyer the second, and the amount of the transaction the third. Both the KWIC and the propositional form are extracted from LN-fr. Finally, the actual question is posed, featuring a new keyword–value pair to be evaluated by the model.

To ensure the reliability of the keyword-value pairs used as query instances, we apply the following sampling constraints when generating prompts: (i) the keyword-value pairs used in the few-shot examples do not appear in the target query; (ii) no duplicate instances are included within the same prompt.

3.4 Evaluation

We evaluated three competitive instruction-tuned LLMs from Transformer (Wolf et al., 2020): QWEN-14B-INSTRUCT-1M (hereafter QWEN), LLAMA-3.1-8B-INSTRUCT (hereafter LLAMA), and MISTRAL-7B-INSTRUCT-V0.3 (hereafter MISTRAL). A total of 81 valid LFs nodes were selected from our classification hierarchy. For each node, we generated 20 questions per contrastive sampling—10 positive ones (based on examples from the target LF) and 10 negative ones (from contrastive LFs)—ensuring a balanced dataset. Each question was posed five times to each model using distinct random seeds, ensuring both reproducibility and the observation of model variance.

In addition, our experimental setup takes into account three parameters, as summarized in Table 1.

Param	Description
k	Number of examples per prompt ($k \in \{2, 6, 10\}$).
`kw-ctx`	Whether the example's keyword includes a KWIC context (boolean, T for True and F for False).
`vl-pfm`	Whether the example's value includes its propositional form (boolean, T for True and F for False).

Table 1: Experimental parameters.

Model	kw-ctx	vl-pfm	k = 2		k = 6		k = 10	
			Acc	F1	Acc	F1	Acc	F1
QWEN	F	F	61.2	59.4	64.6	63.2	66.7	65.7
	F	T	61.5	59.6	65.0	63.9	**67.5**	**66.6**
	T	F	57.9	53.2	62.1	59.1	64.1	62.0
	T	T	58.4	53.6	62.5	59.8	64.6	62.7
LLAMA	F	F	55.7	49.6	58.2	54.3	**59.3**	**55.8**
	F	T	54.5	46.4	56.8	51.4	57.3	52.1
	T	F	54.6	47.5	57.0	52.7	56.7	51.4
	T	T	53.1	43.2	55.0	47.7	54.4	46.4
MISTRAL	F	F	52.5	44.8	53.1	44.0	53.4	44.4
	F	T	53.0	45.8	55.1	48.9	**55.5**	**49.5**
	T	F	50.3	37.0	50.9	37.9	51.6	40.4
	T	T	51.2	40.8	52.6	43.5	52.1	41.5

Table 2: Performance (accuracy and F1 score) of three models under different configurations.

4 Results and discussion

4.1 Global Performance Across Models

General performance overview As shown in Table 2, the overall performance of the three tested models remains relatively modest. Both LLAMA and MISTRAL achieve slightly above the expected accuracy of random guessing in a binary classification task. Even the best-performing model, QWEN, falls short of the 70% threshold, indicating that the lexical relationships involved in this task pose a substantial challenge for these LLMs.

Response polarity bias Given that our evaluation set is strictly balanced, with an equal number of positive ('Yes') and negative ('No') gold labels, any asymmetry in the distribution of predicted labels may reveal a systematic bias in model outputs. As shown in Figure 4, LLAMA and QWEN exhibit a marked preference for predicting 'No', while MISTRAL tends to over-predict 'Yes'. These tendencies suggest distinct response heuristics or inductive biases learned during training, which may influence lexical decision-making in binary setups.

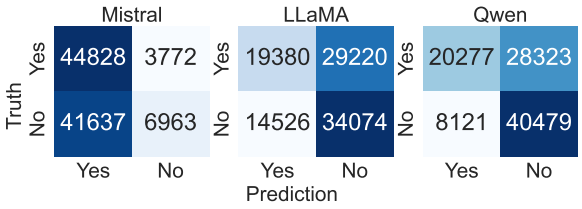


Figure 4: Confusion matrices for the three evaluated models. Rows indicate gold labels, columns show predicted labels. Differences in false positives and false negatives highlight systematic response biases.

4.2 Impact of Experimental Conditions

The three models evaluated in this study exhibit both commonalities and divergences in their performance across experimental conditions. QWEN consistently outperforms the others, followed by LLAMA, with MISTRAL showing comparatively lower accuracy.

Impact of k-shot Table 2 shows that both QWEN and LLAMA demonstrate clear sensitivity to the k-shot instances of target LF provided in the prompt: performance improves steadily as k increases. This suggests that exposure to a greater number of examples enhances the model's ability to recognize and generalize the lexical relation encoded by the target LF. In contrast, MISTRAL's performance remains largely unaffected by changes in k-shot settings, indicating that it may rely less on provided examples into its predictions.

Impact of `kw-ctx` and `vl-pfm` As listed in Table 1, these two parameters are introduced to test their potential role as linguistic cues for disambiguation. However, we observe that none of the three models tested appears to benefit from the inclusion of `kw-ctx`; on the contrary, its presence sometimes leads to even worse performance. On the other hand, `vl-pfm` shows a modest positive effect for both QWEN and MISTRAL, while having little to no impact on LLAMA. It is important to note that the lack of performance improvement from certain prompt components, like `kw-ctx`, does not imply that these types of information are irrelevant to lexical relations. Rather, it indicates that the models, in their current form, fail to effectively leverage such information in making lexical identification.

In the following sections, we focus our subsequent analysis on each model's best-performing configuration (bolded in Table 2), in order to minimize confounding effects from multiple variables.

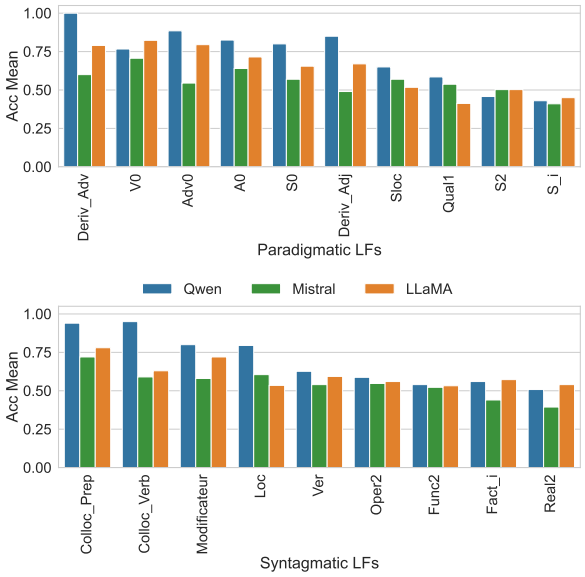


Figure 5: Accuracy of three models (Qwen, LLaMA, Mistral) across selected LFs categories. The upper chart shows performance on a representative set of paradigmatic LFs, while the lower shows performance on syntagmatic LFs.

4.3 Performance Across Lexical Functions

4.3.1 Disparities Among LFs

Since the LFs serve as the central testing material in our task, we begin our analysis by abstracting them away from the hierarchical organization, and examining models performance at the individual LF node. This flat perspective allows us to assess whether the models demonstrate variant accuracy across them.

As illustrated in Figure 5, models generally exhibit accuracy disparities across different target LFs.[3] Some LFs appear easier for the models to learn, particularly when the distinctions are limited to part-of-speech (POS) differences. For instance, `Deriv_Adv` refers to LFs that, given a lexical unit as the keyword, return an adverbial lexical unit derived from it while preserving the semantics, and it is contrasted with other derivations (nominal, adjectival, etc.) as counter examples. The results suggests that, when prompted to decide whether a pair such as (*rapide* 'rapid', *rapidement* 'rapidly') fits this pattern, models often respond with high accuracy, with QWEN even hitting perfect scores on this LF with some configurations.

Conversely, some LFs are considerably more challenging for the models, particularly when they involve semantic argument structures. For example, $\texttt{Func}_2$ is defined as an LF that, given a non-verbal keyword, returns a support verb, allowing to build a construction that functions as a verb without altering the meaning of the keyword, and in this structure, the keyword functions as the subject of the verb, and its semantic argument 2 becomes the direct object of the verb. For example, $\texttt{Func}_2(\textit{blow}_{\text{N}})$ returns $\textit{fall}_{\text{V}}$ as seen in the collocation *the blow falls upon* y. In our experiments, $\texttt{Func}_2$ is contrasted with $\texttt{Func}_0$ (e.g. $\texttt{Func}_0(\textit{silence}_{\text{N}}) = \textit{reign}_{\text{V}}$), which shares the same syntactic and semantic properties but lacks an additional argument serving as the verb's object, and $\texttt{Func}_1$ (e.g. $\texttt{Func}_1(\textit{blow}_{\text{N}}) = \textit{come}$—as in *the blow comes from* x), in which the keyword's semantic argument 1 becomes the direct object of the verb.[4] The models consistently struggle to distinguish such nuanced semantico-syntactic patterns, with performance occasionally dropping below random level.

This disparity is similar to the observation in the ALF study (Petrov et al., 2025) and suggests that LLMs have varying degrees of understanding across different types of LFs. Below, we examine whether these disparities may be shaped by our hierarchical organization of LFs (cf. §3.1).

4.3.2 Hierarchical Patterns in LF-Specific Performance

To gain deeper insight into the observed disparities (§4.3.1), we regroup all LFs based on their depth in the hierarchy (cf. Figure 1) and analyze how model performance varies across different levels of abstraction.

As illustrated in Figure 6, models indeed demonstrate systematic performance disparities in performance across LFs by their depth levels. For both QWEN and LLAMA, deeper LFs—which denote more specific distinctions—are associated with greater classification difficulty, with QWEN displaying a particularly marked decline. While MISTRAL exhibits a certain degree of insensitivity to depth at higher hierarchical levels, substantial decline in accuracy is evident at the lowest tiers of the structure. By linking these depth levels in the hierarchy to the disparities introduced earlier, we find that LFs associated with clearer distinctions in part-of-speech—such as `Deriv_Adv`—correspond to the top-level (depth = 1), where models generally

[3] Figure 5 illustrates a representative sample from the full set of 81 targets.

[4] See Mel'čuk and Polguère (2021); Mel'čuk (1996) for a comprehensive overview of these LFs.

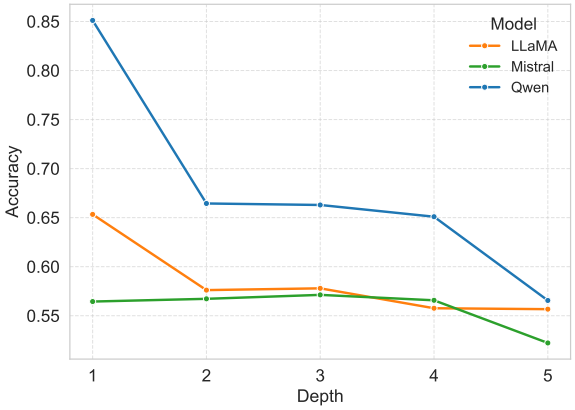


Figure 6: Performance trends across lexical functions grouped by their depth in the hierarchical classification (with 1 denoting the top-level LF nodes and 5 denoting the most fine-grained nodes). Each curve represents one model's average performance on target LFs at a given depth in the hierarchy, measured by accuracy.

LF group	QWEN	LLAMA	MISTRAL	Mean
$S_1, S_2, \ldots$	0.58	0.57	0.53	0.56
$S_{res}, S_{loc}, \ldots$	0.76	0.63	0.58	0.65

Table 3: Example of performance contrast: S_i (with indices referring to arguments) versus S_{res}, etc. (without such indices)

perform well. In contrast, more challenging LFs such as $Func_2$ are situated deeper in the hierarchy, where classifications become more fine-grained. This observation supports our earlier hypothesis that disparities in model performance are partially shaped by the hierarchical organization of LFs.

4.3.3 Challenges of Argument-Aware LFs

While hierarchical depth plays an important role in shaping performance differences, we also observe another layer of complexity arising from the argument structures encoded in certain LFs. One plausible explanation lies in the conventional, rather than absolute, nature of semantic arguments: their interpretation often depends on norms among linguists rather than fixed rules. For instance, S_1 and S_2, introduced in §3.1, belong to LFs that refer to the argument structure of the keyword. However, when compared to nodes like S_{instr} or S_{res} at similar depths without argument indices, model performance varies considerably, despite their similar hierarchical depth.

As contrasted in Table 3, LFs characterized by clearer semantic interpretations—without reliance on semantic argument numbers—tend to be more consistently recognized. This may help explain why the `vl-pfm` parameter improves accuracy for models like QWEN and MISTRAL, as it provides disambiguating signals that compensate for such variability.

4.4 Impact of Morphological Similarity between Keywords and Values

Semantic and syntactic relations form the core of the LFs linking two lexical units. In French LF examples, however, these relations are often accompanied by morphological similarity between the keyword and its value. To assess whether models rely on surface-form resemblance rather than structural understanding of LFs, we measured the similarity of pair of words using scores between word pairs using the `Levenshtein_ratio()` function from the python-Levenshtein library.[5] Unlike the raw *Levenshtein distance* (Levenshtein, 1966) which counts the minimum number of single-character edits needed to transform one string into another, this function returns a normalized similarity score between 0 and 1, providing a convenient proxy for morphological relatedness.

4.4.1 Correlation between Morphological Similarity and Models' Responses

We first hypothesize that models' responses (*Yes*/*No*) may be influenced by the morphological similarity of the *keyword-value* pair in posed questions; higher similarity might bias the model toward a specific polarity. To delve into this inquiry, we measured the correlation between the morphological similarity of each *keyword–value* pair and the response polarity (*Yes*/*No*) using the *Pearson Correlation Coefficient*. The results, shown in Figure 7, reveal that this correlation varies across LFs too.

Model	A_0	Contr	Pred	V_0
LLAMA	0.70	0.52	0.66	0.79
MISTRAL	0.63	0.42	0.46	0.76
QWEN	0.80	0.40	0.90	0.74

Table 4: Accuracy scores for selected lexical functions across models.

For V_0 (e.g., V_0(*driving*$_N$)=*drive*$_V$), the high positive value in the light-red bar indicates that higher pair similarity is associated with *Yes* answers; all 3 tested LLMs align to varying degree to

[5] https://github.com/ztane/python-Levenshtein

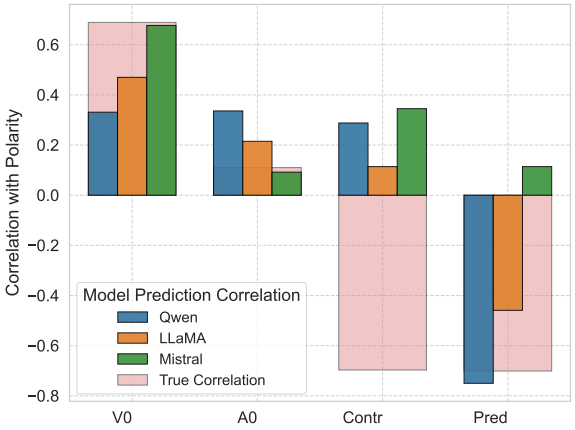


Figure 7: Correlation between the morphological similarity of each *keyword–value* pair and the response polarity (*yes*/*no*). The light-red background shows the correlation of this similarity with ground-truth response, and the colored bars (blue, orange, green) show the correlation with each LLM's predictions. As the polarity (*yes*/*no*) was binarized to +1 and −1, values close to +1 indicate that higher similarity is associated with *yes* responses, values close to −1 indicate association with *no* responses, and values near 0 indicate little correlation.

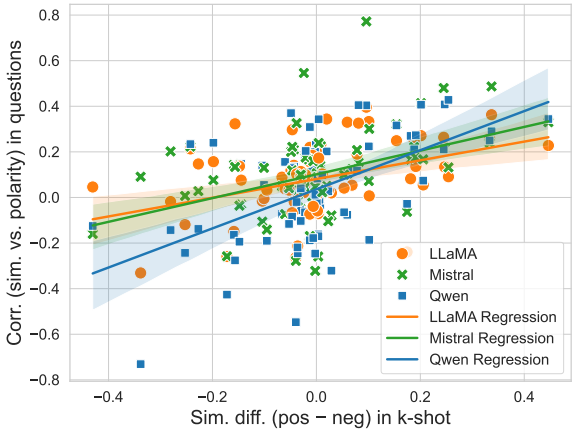


Figure 8: For each LF, relationship between (i) the correlation of question-pair similarity with answer polarity (defined in Section §4.4.1) and (ii) the difference between the average similarity of positive and negative k-shot examples (positive values indicate higher similarity for positives). Each point represents one LF; colors denote models, and regression lines show the fitted relationship for each model.

this trend. When a model's prediction correlation matches the ground truth, it suggests reliance on surface similarity, often with higher accuracy (see Table 4). For `Pred` (e.g., `Pred`(*beer*)=*drink*$_V$), the negative value indicates that similarity is more associated with *No* answers; QWEN aligns and performs best, while MISTRAL shows no such alignment and performs worst. For `Contr` (e.g., `Contr`(*sun*)=*moon*), none of the models align with the ground truth, and overall performance is weak. These observations suggest that the evaluated LLMs do make use of morphological similarity as a cue for inference, but in ways that vary across LFs.

4.4.2 Prompt Contrast as a Source of Similarity Bias

LLMs' reliance on morphological similarity, as observed in Section §4.4.1 was limited to the *keyword-value* pairs in the questions, we further explore whether this reliance may also be related to the pairs in positive and negative examples (k-shot). For each LF, we first computed the correlation between question-pair similarity and the model's predictions (as defined in the previous section), and then calculated the difference between the average similarity of positive and negative examples in its k-shot context. Figure 8 visualizes the relationship between these per-LF correlations and similarity difference.

All three regression lines have a clear upward slope, supporting our hypothesis that when positive examples are more similar than negative ones, models tend to answer *yes*; the opposite pattern leads to *no*. Notably, MISTRAL shows a shallower slope, whereas QWEN's is steeper, suggesting that QWEN is relatively more capable of capturing the morphological similarity contrast between positive and negative examples in the k-shot and using it to guide its *Yes*/*No* responses. The relative ordering of the slopes aligns with their global performance reported earlier in §4.1.

5 Conclusion

In this study, we introduce a structured benchmark for evaluating LLMs' lexical competence, grounded in a semantic–syntactic hierarchical classification of LFs. Using contrastive prompts, we find that models can leverage lexical cues but struggle with deeper distinctions. They perform better on surface-level PoS contrasts, while finer-grained or syntactically nuanced LFs pose greater challenges. Moreover, model responses are partly driven by morphological similarity between word pairs, especially when such cues are amplified by the prompt design.

Limitations

Our present evaluation is restricted to three mid-sized open-weight LLMs, and we plan to extend the benchmark to larger and more diverse models. In addition, the LF classification follows a semantics-to-syntax ordering which, while theoretically grounded, may not reflect alternative organizational perspectives; exploring alternative LF classifications could help assess structural effects. Furthermore, human evaluation—both with participants familiar and unfamiliar with LF theory—could serve as a valuable baseline for comparing LLM performance; yet this approach has not been widely tested with human participants. In this regard, Petrov et al. (2025) offer a useful point of reference.

Acknowledgments

This research was funded by the Social Sciences and Humanities Research Council of Canada (RNH02072) and the Fonds de recherche du Québec (366841).

References

Margarita Alonso Ramos. 2015. El diccionario de colocaciones del español: Una puesta al día. *Estudios de lexicografía*, 5:103–122.

Juri Apresjan. 2000. *Systematic Lexicography*. Oxford University Press.

ATILF. 2024. French lexical network (fr-ln). ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr.

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, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report.

María Auxiliadora Barrios Rodríguez. 2024. Diretes, a spanish monolingual dictionary based on lexical-semantic relations. In *Lexicography and Semantics. Proceedings of the XXI EURALEX International Congress*, pages 393–407, Cavtat. Institut za hrvatski jezik.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2023. A survey on evaluation of large language models.

Jiangjie Chen, Rui Xu, Ziquan Fu, Wei Shi, Zhongqiao Li, Xinbo Zhang, Changzhi Sun, Lei Li, Yanghua Xiao, and Hao Zhou. 2022. E-KAR: A benchmark for rationalizing natural language analogical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3941–3955, Dublin, Ireland. 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.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15, San Diego, California. Association for Computational Linguistics.

Anne-Laure Jousse. 2010. *Modèle de structuration des relations lexicales basé sur le formalisme des fonctions lexicales*. Ph.D. thesis, Université de Montréal & Université Paris 7, Montréal/Paris.

Vladimir I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.

Marie-Claude L'Homme, Marie-Ève Laneville, and Daphnée Azoulay. 2009. Le dictionnaire fondamental de l'environnement. Technical report.

Veronika Lux-Pogodalla and Alain Polguère. 2011. Construction of a French Lexical Network: Methodological Issues. In *Proceedings of the First InternationalWorkshop on Lexical Resources, WoLeR 2011. An ESSLLI 2011 Workshop*, pages 54–61, Ljubljana, Slovenia.

Mathieu Mangeot. 2000. Papillon Lexical Database Project: Monolingual Dictionaries and Interlingual Links. In *WAINS'7, 7th Workshop on Advanced Information Network and System*, page 6, Kasetsart University, Bangkok, Thailand.

Igor A. Mel'čuk. 1973. Towards a linguistic 'meaning-text' model. In F. Kiefer, editor, *Trends in Soviet Theoretical Linguistics*, pages 33–57. Springer Netherlands, Dordrecht.

Igor A. Mel'čuk. 1996. Lexical functions: A tool for the description of lexical relations in a lexicon. In Leo Wanner, editor, *Lexical Functions in Lexicography and Natural Language Processing*, pages 37–103. Benjamins, Amsterdam/Philadelphia.

Igor A. Mel'čuk. 2016. *Language: From Meaning to Text*. Academic Studies Press, Boston.

Igor A. Mel'čuk, André Clas, and Alain Polguère. 1995. *Introduction à la Lexicologie Explicative et Combinatoire*. Duculot, Louvain-la-Neuve.

Igor A. Mel'čuk and Alain Polguère. 2021. Les fonctions lexicales dernier cri. In Sébastien Marengo, editor, *La théorie Sens-Texte. Concepts-clés et applications*, pages 75–155. L'Harmattan, Paris.

Igor A. Mel'čuk, Nadia Arbatchewsky-Jumarie, Lidija Iordanskaja, Suzanne Mantha, Alain Polguère, and André Clas. 1999. *Dictionnaire explicatif et combinatoire du français contemporain. Recherches lexico-sémantiques IV: Recherches lexico-sémantiques IV*. Presses de l'Université de Montréal.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *ACL 2022 - 60th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, pages 3470–3487. Association for Computational Linguistics (ACL).

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, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Alexander Petrov, Antoine Venant, François Lareau, Yves Lepage, and Philippe Langlais. 2025. ALF: Un jeu de données d'analogies françaises à grain fin pour l'évaluation de la connaissance lexicale des grands modèles de langue. In *Actes de la 32e conférence sur le traitement automatique des langues naturelles (TALN)*, volume 1, pages 22–49, Marseille, France.

Alain Polguère. 2014. From writing dictionaries to weaving lexical networks. *International Journal of Lexicography*, 27(4):396–418.

Margarita A. Ramos and Agnès Tutin. 1996. A classification and description of lexical functions for the analysis of their combinations. In Leo Wanner, editor, *Lexical Functions in Lexicography and Natural Language Processing*, pages 147–167. Benjamins, Amsterdam/Philadelphia.

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

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.

Peter Turney, Michael Littman, Jeffrey Bigham, and Victor Shnayder. 2004. Combining independent modules in lexical multiple-choice problems. In *Recent Advances in Natural Language Processing III: Selected papers from RANLP 2003*, pages 101–110.

Asahi Ushio, Luis Espinosa Anke, Steven Schockaert, and Jose Camacho-Collados. 2021. BERT is to NLP what AlexNet is to CV: Can pre-trained language models identify analogies? 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 3609–3624, Online. Association for Computational Linguistics.

Thilini Wijesiriwardene, Ruwan Wickramarachchi, Bimal G. Gajera, Shreeyash Mukul Gowaikar, Chandan Gupta, Aman Chadha, Aishwarya Naresh Reganti, Amit Sheth, and Amitava Das. 2023. Analogical – a novel benchmark for long text analogy evaluation in large language models.

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.

Siyu Yuan, Jiangjie Chen, Changzhi Sun, Jiaqing Liang, Yanghua Xiao, and Deqing Yang. 2024. ANALOGYKB: Unlocking analogical reasoning of language models with a million-scale knowledge base. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1249–1265, Bangkok, Thailand. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2025. A survey of large language models.

A German WSC dataset comparing coreference resolution by humans and machines

Wiebke Petersen
Institute of Linguistics
Heinrich-Heine Universität
Düsseldorf, Germany
wiebke.petersen@hhu.de

Katharina Spalek
Institute of Linguistics
Heinrich-Heine Universität
Düsseldorf, Germany
katharina.spalek@hhu.de

Abstract

We present a novel German Winograd-style dataset for direct comparison of human and model behavior in coreference resolution. Ten participants per item provided accuracy, confidence ratings, and response times. Unlike classic WSC tasks, humans select among three pronouns rather than between two potential antecedents, increasing task difficulty. While majority vote accuracy is high, individual responses reveal that not all items are trivial and that variability is obscured by aggregation. Pretrained language models evaluated without fine-tuning show clear performance gaps, yet their accuracy and confidence scores correlate notably with human data, mirroring certain patterns of human uncertainty and error. Dataset-specific limitations, including pragmatic reinterpretations and imbalanced pronoun distributions, highlight the importance of high-quality, balanced resources for advancing computational and cognitive models of coreference resolution.

1 Introduction

Coreference resolution is a central task in NLP (for a review see Zhang et al., 2021), with most work focusing on fine-tuning models for benchmark performance (e.g., Wang et al., 2019). In contrast, we directly compare the behavior of humans and pretrained language models (PTLMs) on a task requiring coreference resolution. Prior work shows that PTLMs encode coreference-relevant biases – such as preference for form similarity, recency, and grammatical agreement – when probed via contextual embeddings (Sorodoc et al., 2020), mirroring patterns found in human anaphora resolution (e.g., Ariel, 2001; Stevenson et al., 1995). Yet for direct human–machine comparison, analyzing PTLM behavior during sentence processing offers more insight than diagnostic probing. Following Ettinger (2020), we therefore assess PTLMs in a psycholinguistic setup.

We investigate how humans and PTLMs process German Winograd Schemas coreference problems designed to test commonsense reasoning and named after an example in Winograd (1972). The Winograd Schema Challenge (WSC) (Levesque et al., 2012) was proposed as a more demanding alternative to the Turing Test (Turing, 1950).[1] WSC items involve ambiguous pronouns whose resolution requires commonsense reasoning, and they are generally regarded as easy for humans but difficult for machines. A classic Winograd Schema (WS) consists of a pair of sentences differing only in a single critical word, that flips the intended referent of the pronoun. The classic task was to identify the correct antecedent:

(1) Jane gave **Joan** candy because **she** was hungry.
Jane gave Joan candy because **she** wasn't hungry.
Who [was/wasn't] hungry? [( ) Jane; ( ) Joan]

In parallel experiments, we compare how humans and machines differ in processing coreference. Specifically, we investigate (1) whether the same items are perceived as difficult by both groups, (2) which group performs better overall, and (3) whether model-based confidence measures (e.g., softmax probabilities) align with human self-assessed confidence ratings or response times. To ensure comparability, both groups perform *the same task on the same data*. Since we are interested in the linguistic knowledge encoded by pretrained models rather than in their capacity for fine-tuning, we deliberately refrain from additional training. Instead, we construct a dedicated dataset that allows

[1] The original Turing Test (judging whether a conversational partner is human or not) has been criticized as too easy to pass through shallow mimicry rather than genuine understanding (Weizenbaum, 1966).

Proceedings of the 16th International Conference on Computational Semantics, pages 120–127
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

direct human–machine comparison on items reflecting tasks already encountered during pretraining.

2 Experiments

The key idea of our approach is to directly compare human and machine behavior on coreference resolution using a cloze-style task in German based on WSC items (see Fig. 1 for an example). The original WSC dataset (WSC273)[2] comprises 273 manually constructed WSC pairs like those in (1), where the task is to choose between two potential antecedents. The pairs are designed to meet three criteria: (a) the correct referent is unambiguous for humans; (b) resolution cannot rely solely on selectional restrictions; and (c) frequency-based heuristics are insufficient. Due to their difficulty and significance for machine translation and anaphora resolution, several larger WSC-style datasets have since been created. Among them, WinoGrande (Sakaguchi et al., 2021) is the most prominent, containing around 44,000 sentence pairs developed and validated via crowdsourcing. These are presented in cloze format, with the ambiguous pronoun replaced by a blank to be filled in, and two candidate antecedents provided as answer options. Reported human accuracy on these datasets typically exceeds 90% or even 95% (Kocijan et al., 2023).

Our approach is related to Abdou et al. (2020), who tested the robustness of humans and PTLMs on perturbed WSC items in cloze format (e.g., voice or tense changes), comparing majority vote (humans) and softmax predictions (PTLMs). While they focused on accuracy and stability, we go further by comparing confidence levels. To ensure comparability between the human and machine experiments we (1) avoid task priming by using fillers (humans) and no fine-tuning (machines), and (2) present structurally identical items to both groups.

2.1 Materials and task

We curated a set of 50 German WSC pairs satisfying two conditions: (i) each sentence contains two singular noun phrases of different grammatical gender and a gap to be filled with a nominative singular pronoun; (ii) both sentences differ in one critical word that determines the correct referent. Twenty-five pairs were randomly drawn from the `lm_en_de` subset of MT-Wino-X (Emelin and Sennrich, 2021, here: Wino-X), a multilingual extension of WinoGrande for machine translation. The remaining 25 were translated from WSC273 using DeepL and manually revised to ensure grammaticality, naturalness, and a nominative singular pronoun gap. In cases where the gender of the two candidate referents did not differ, we adapted the referents accordingly (see (2), based on (1); for more details on the data adaption process see Appendix A).

(2) Jan_{masc} gab Anne_{fem} Süßigkeiten, weil ___ *satt/hungrig* war.

The final dataset comprises 100 Winograd items (50 pairs): 42 with ‘sie’ (she) as the gold answer, 39 with ‘er’ (he), and 19 with ‘es’ (it). It was used in both experiments (humans and machines).

2.2 Human behavioral experiment

Using the dataset described in 2.1, we created ten experimental lists. Each list contained ten different WSC items (five from Wino-X, five from translated WSC273) and fifteen filler items, each presented as a cloze task (see Fig. 1). For no WSC pair both items belong to the same list. Filler items were designed to obscure the logical structure of the WSC-problems. As fillers we used sentences with only one potential antecedent, including entities with fixed (grammatical) gender (e.g., ‘der Tisch’) and ambiguous gender (e.g., brand names like ‘Nutella’, proper names like ‘Alex’ or foreign words like ‘Laptop’).

Participants selected the fitting German pronoun (*er*, *sie*, *es*) for each gap and rated their confidence on a 1–5 scale (see Fig. 1). Reaction times were recorded for both decisions. We tested 100 native German speakers (aged 18–55), collecting ten responses per WSC item. The experiment was conducted online using PsychoPy (Peirce, 2007) and distributed with via Clickworker. The experiment took 10-15 minutes, and participants received a small monetary compensation of 2.50€.

2.3 Pretrained language models (PTLMs) behavioral experiment

Our goal is to compare human and machine behavior on WSC items as directly as possible. We therefore evaluate PTLMs on the same cloze-style tasks used in the human experiment, without task-specific fine-tuning. This allows us to assess their inherent capabilities for coreference resolution based solely on their masked language modeling (MLM) pretraining.

[2] https://www.tensorflow.org/datasets/catalog/wsc273

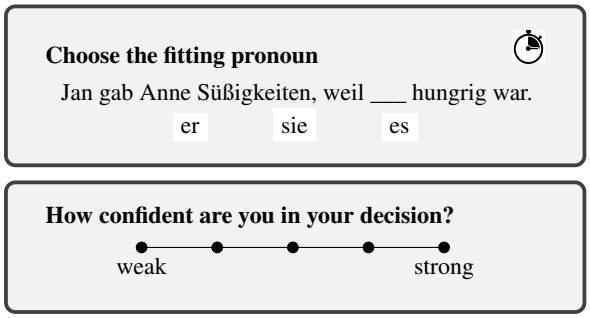


Figure 1: Human behavioral experiment: Pronoun choice and confidence rating, presented at two consecutive screens.

We include three BERT-based models: `bert-base-german-cased`, `gbert-large` (Chan et al., 2020), and `xlm-roberta-large` (Conneau et al., 2020). Each WSC item is converted into fill-mask format using the appropriate mask token. Softmax-normalized scores over the token vocabulary are interpreted as the model's confidence in a token being the correct filler.

A key challenge is the mismatch between the tasks: humans are forced to choose one of three given pronouns (*er*, *sie*, *es*), while PTLMs predict freely from the entire vocabulary. To address this, we implement three configurations:

In the **pron**-configuration, only the three target pronouns are considered. The highest-scoring token among *er*, *sie*, and *es* defines the model's prediction and its confidence. The score for the gold answer serves as the target confidence. This setup, however, disregards other high-scoring tokens that may function as pronoun synonyms in context.

The **topk**-configurations approximate the human task by including pronoun variants. The model's top-k predictions are mapped to gendered pronoun classes (*masc*, *fem*, *neut*, *other*) using curated lists.[3] The softmax scores of all top-k tokens belonging to each class are summed; the class with the highest total defines the model's prediction and its confidence. We test $k = 10$ and $k = 1$, using either the summed gold-class score (**top10**) or the top-scoring gold-class token (**top1**) as target confidence.

Each configuration yields: (i) the model's prediction, (ii) its correctness, (iii) its confidence in its given answer, and (iv) its target confidence (i.e., how strongly it favors the gold answer).

[3]E.g., *masc*: *der*, *er*, *dieser*, *jener*, etc.

3 Results and Discussion

In the behavioral experiment, we find a moderate inter-annotator agreement among humans (κ = 0.562), with only 28 items answered unanimously. This relatively low agreement is itself an important finding. First, it challenges the common assumption that WSC items are straightforward for humans and thus constitute a reliable benchmark for evaluating machines. Second, it raises concerns about the widespread practice of defining the human "gold" response via majority vote from as few as three annotators per item (see Kocijan et al., 2023, for a survey). The observed lack of high inter-annotator agreement suggests that majority votes based on larger samples may yield substantially different outcomes. Notably, for 21 items all three pronouns were chosen by at least one participant. At the same time, high agreement ($\geq$7 of 10 participants selecting the same pronoun) was reached for 82 items, showing that while some items elicited highly consistent responses, a substantial number provoked genuinely ambiguous interpretations.

Table 1 summarizes model and human performance on our referential pronoun resolution task. Among models, `GBERT-large` and `XLM-RoBERTa-large` perform comparably (accuracy $\approx$ 0.56), both outperforming the smaller `bert-base-german-cased` (accuracy $\approx$ 0.53). Between configurations, accuracy remains largely stable, with `top10` showing the highest target confidence, closely followed by `pron`, while `top1` exhibits a notable drop. `XLM-RoBERTa-large` achieves slightly higher target confidence than `GBERT-large` and is therefore used in subsequent analyses. Overall, model accuracy is somewhat lower than previously reported results on English WSC data (~60%, Kocijan et al., 2023), likely due to the increased complexity of our task: models perform *free-form generation* over the full token vocabulary and humans choose among *three* options (rather than two in classic WSC).

Human performance is considerably higher than model performance, but also reveals striking variability. While majority vote accuracy is relatively high (0.87), individual accuracy is markedly lower (mean = 0.729). This challenges the assumption that WSC-style problems are trivial for humans and highlight the limitations of majority-based metrics, potentially masking individual uncertainty.

A breakdown by dataset reveals a strong quality gap: performance on Winograd-style expert-

Model	Accuracy			Target Conf.		
	top1	top10	pron	top1	top10	pron
XLM-RoBERTa	0.56	0.57	0.55	0.411	0.495	0.414
GBERT-large	0.56	0.56	0.56	0.397	0.481	0.410
BERT-base-german	0.53	0.52	0.53	0.342	0.412	0.350
`Human (indiv.)`	0.729			–		
`Human (majority)`	0.870			–		

Table 1: Model and human performance on Winograd cloze tasks. Accuracy refers to the proportion of correct predictions. Target confidence corresponds to the softmax score assigned to the gold token.

Model conf.	Acc.	Rating	RT
top1 (gold)	0.418	0.263	-0.245
top10 (gold)	0.410	0.280	-0.247
pron (gold)	0.486	0.307	-0.310
top1 (given)	0.222	0.260	-0.214
top10 (given)	0.120	0.225	-0.127
pron (given)	0.287	0.360	-0.253

Table 2: Pearson correlations between model confidence scores in gold and given answer and human measures (Acc. = correlation with mean human accuracy, Rating = correlation with mean human confidence ratings, RT = correlation with mean human reaction times). All p-values < 0.001.

Model config.	V (indiv.)	V (maj. vote)
top1	0.433	0.503
top10	0.428	0.520
pron	0.441	0.503

Table 3: Cramér's V between model predictions and human responses (individual and majority vote).

curated items (WSC273) is substantially higher than on Wino-X items, which are based on crowd-generated and machine translated data. Human majority vote accuracy is perfect on WSC273 but drops to 0.74 on Wino-X. Individual accuracy follows the same pattern (0.85 vs. 0.61). Model performance mirrors this trend (`pron`: 0.60 vs. 0.50), underscoring the importance of data curation.

In our analysis of human behavioral correlations, items with lower mean accuracy elicited longer mean response times ($r = -0.194$, $p < .001$) and lower mean confidence ratings ($r = 0.256$, $p < .001$). For the most extreme deciles mean accuracy raises from 0.65 for the slowest 10% of responses to 0.71 for the fastest, and from 0.49 for the lowest-rated items to 0.79 for the highest-rated ones, reinforcing the validity of these behavioral metrics. At the participant level, response time and confidence are themselves negatively correlated ($r = -0.142$, $p < .001$), indicating that individuals tended to take longer when less certain.

Comparing human and model behavior, we first note that all models predict the human majority vote more accurately than the gold answer (Table 5 in the appendix vs. Table 1). This suggests that models partially mirror human error patterns and produce judgments that align with aggregated human preferences.

Correlations between model confidence (measured as softmax scores for both the gold and given answer) and human behavioral measures are shown in Table 2. Unsurprisingly, due to the higher accuracy of human answers, model confidence in the gold answer correlates more strongly with human accuracy than confidence in the given answer. The same pattern is observed for correlations with human confidence ratings and response times, although the difference between gold and given answer is much smaller in this case. While our original aim was to approximate model confidence via the given answer, gold-answer confidence ultimately shows the closest alignment with human behavior. Finally, despite its weak accuracy (see Table 1), the `pron` configuration shows the strongest correlations with human data across all configurations. Thus, despite lower correctness, its confidence estimates align more closely with human uncertainty and difficulty.

Complementing this, Cramér's V analysis (Cramér, 1946) reveals moderate alignment between model outputs and individual human responses (Table 3). Again, `pron` shows the highest similarity to individual human response patterns, underscoring its ability to capture human-like behavior despite lower correctness. Additionally, model confidence in given answer correlates positively with human agreement (for `pron`: Spearman $r = 0.359$, $p < .001$; see Fig. 2 in the appendix), indicating that models are more confident when humans agree.

We observe several notable human and model error patterns. First, humans frequently select *es* to refer to an entire situation rather than one of the intended antecedents. Excluding items with wrong majority vote *es*, majority vote accuracy rises to 0.97 and individual accuracy to 0.80, indicating that many apparent 'errors' are due to pragmatic reinterpretation (see Appendix B for examples).

Second, models show markedly less variability

across paired items. While humans gave identical answers to both items of a pair in 10 cases, models did so over 35 times (e.g., 37 in pron). This suggests a tendency to being biased and ignore subtle contextual shifts that differentiate minimal pairs.

Third, both humans and models exhibit systematic biases in antecedent selection.[4] Human responses show a slight preference for the *first antecedent* (515 vs. 424), while models exhibit a stronger bias toward the *second antecedent* (e.g., pron: 57 vs. 41), reflecting the recency bias observed in probing studies (Sorodoc et al., 2020). Model accuracy is higher when the correct referent is the first antecedent (e.g., pron: 0.59 vs. 0.54), while humans perform better when the correct answer is in the second position (0.74 vs. 0.81). This asymmetry is challenging to interpret, as pronoun types are not evenly distributed across positions and gender biases may influence performance. For instance, models perform best on *es* (pron: 0.68), followed by *sie* (0.57) and *er* (0.46), while humans show a minor difference between *er* (0.72) and *sie* (0.71), and a pronounced advantage for *es* (0.80).

4 Conclusion

We presented a novel German Winograd-style dataset and collected fine-grained human data, including accuracy, confidence ratings, and response times, with 10 participants per item.[5] This resource provides a rich empirical basis for studying referential resolution in German and evaluating model behavior. Thereby our task setup is more challenging than previous WSC formulations: humans must choose among three pronouns, and models face open-ended generation over the entire vocabulary. Despite this, our results show clear human–machine performance gaps, alongside intriguing similarities in uncertainty and error patterns.

At the same time, our analysis reveals limitations in the dataset itself: some 'errors' reflect pragmatic reinterpretations rather than misunderstanding. Moreover, pronoun distribution is uneven across antecedent positions, suggesting room for improvement in future dataset design. Taken together, our findings reinforce the critical importance of high-quality, carefully constructed data for both cognitive and computational modeling of reference resolution.

[4]Note that the WSC pairs are balanced such that each antecedent position is correct equally often.

[5]The dataset is available from the authors upon request.

References

Mostafa Abdou, Vinit Ravishankar, Maria Barrett, Yonatan Belinkov, Desmond Elliott, and Anders Søgaard. 2020. The sensitivity of language models and humans to Winograd schema perturbations. In *Proc. of the 58th Annual Meeting of the ACL*, pages 7590–7604. ACL.

Mira Ariel. 2001. Accessibility theory: An overview. *Text Representation: Linguistic and Psycholinguistic Aspects*, 8(8):29–88.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on 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.

Harald Cramér. 1946. *Mathematical Methods of Statistics*, volume 9 of *Princeton Mathematical Series*. Princeton University Press.

Denis Emelin and Rico Sennrich. 2021. Wino-X: Multilingual Winograd schemas for commonsense reasoning and coreference resolution. In *Proc. of the 2021 EMNLP*, pages 8517–8532, Punta Cana, Dominican Republic. ACL.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *TACL*, 8:34–48.

Vid Kocijan, Ernest Davis, Thomas Lukasiewicz, Gary Marcus, and Leora Morgenstern. 2023. The defeat of the Winograd schema challenge. *Artificial Intelligence*, 325:103971.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd Schema Challenge. In *Proc. of the 13th International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, pages 552–561. AAAI Press.

Jonathan W. Peirce. 2007. PsychoPy—Psychophysics software in Python. *Journal of Neuroscience Methods*, 162(1-2):8–13.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial Winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Ionut-Teodor Sorodoc, Kristina Gulordava, and Gemma Boleda. 2020. Probing for Referential Information

in Language Models. In *Proc. of the 58th Annual Meeting of the ACL*. ACL.

Rosemary J Stevenson, Alexander WR Nelson, and Keith Stenning. 1995. The role of parallelism in strategies of pronoun comprehension. *Language and Speech*, 38(4):393–418.

A. M. Turing. 1950. Computing machinery and intelligence. *Mind*, LIX(236):433–460.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Neural Information Processing Systems*, pages 3266–3280.

Joseph Weizenbaum. 1966. ELIZA – a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

Terry Winograd. 1972. Understanding natural language. *Cognitive Psychology*, 3(1):1–191.

Hongming Zhang, Xinran Zhao, and Yangqiu Song. 2021. A brief survey and comparative study of recent development of pronoun coreference resolution in English. In *Proc. of the Fourth Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 1–11, Punta Cana, Dominican Republic. ACL.

A Details: Adaption of German WSC items for experiments

Half of the WSC items used in our experiments were drawn from the Wino-X dataset; the other half are adaptations of items from the original WSC273 set.

The `lm_en_de` subset of Wino-X is a subset of WinoGrande containing the English pronoun 'it'. These were automatically translated into German, with 'it' replaced by a gap. Sentence pairs in which both versions required a pronoun of the same grammatical gender in German were excluded. For our study, we randomly selected 25 sentence pairs from this subset, ensuring only that the blank required a nominative singular pronoun. No further manual filtering or quality control was applied.

For the remaining 25 items, we randomly selected examples from WSC273 and translated them into German using DeepL. We then replaced the pronoun position with a blank and manually adjusted the sentences to (i) ensure grammatical fluency, (ii) require a nominative singular pronoun, and (iii) introduce two potential antecedents with different grammatical genders.

An example adapted from an original WSC pair is shown below:

(3) a. original: The firemen arrived before the police because they were coming from so far away.

b. German adaption: Der Krankenwagen$_{\text{masc}}$ kam vor der Polizei$_{\text{fem}}$, weil ___ so einen weiten Weg hatte.
The ambulance came before the police because ___ had such a long way.

(4) a. original: The firemen arrived after the police because they were coming from so far away.

b. German adaption: Der Krankenwagen$_{\text{masc}}$ kam nach der Polizei$_{\text{fem}}$, weil ___ so einen weiten Weg hatte.
The ambulance came before the police because ___ had such a long way.

The original English pair used the plural pronoun 'they', which was incompatible with our singular-pronoun setup. The automatic DeepL translation rendered the feminine singular nouns 'fire department' ('Feuerwehr') and 'police' ('Polizei'). To introduce a gender contrast, we replaced 'Feuerwehr' with 'Krankenwagen' ('ambulance', masculine), enabling unambiguous pronoun resolution.

All item adaptions followed a similar procedure. Plural noun phrases were converted to singular, and gender-specific alternatives were introduced where necessary. Original WSC273 pairs typically involved ambiguous pronouns and names matched for gender. To ensure disambiguation via grammatical gender in German, we replaced personal names with frequent German names stereotypically associated with different genders.

B Human Majority Vote Errors

We begin by examining those 13 WSC items (out of 100) where the human majority vote diverged from the expected response. These instances highlight potential flaws in the item design, calling into question the claim that WSC-style problems are straightforward for humans. Fig. 2 shows that wrong majority votes occur across all agreement levels.

Several sources of confusion were identified:
Perspective shift: Some items allow for both pronouns to result in a coherent sentence by shifting

the perspective on the critical word.

(5) "Die Frau kaufte eine Muschel$_{fem}$, um sie ins Aquarium$_{neu}$ zu stellen, weil ___ schlicht aussah."
majority vote: *sie* expected response: *es*
The woman bought a shell to put into the aquarium because ___ looked plain.

Both interpretations are plausible: either the shell is plain (*sie*) – and the women likes plain and simple things – or the aquarium is perceived as looking too plain without it (*es*).

Situational reference: Frequently, participants chose *es* to refer not to a noun, but to the entire situation. 50 times a participant answered *es* although neither the first nor the second antecedent had neuter gender.

(6) Clara beschloss, Gemüse im Ofen$_{masc}$ anstatt in der Mikrowelle$_{fem}$ zu kochen, weil ___ das Gemüse saftiger schmecken ließ.
majority vote: *es* expected response: *er*
Clara decided to cook vegetables in the oven rather than the microwave because ___ made them taste juicier.

Here, *es* refers to the preparation process rather than a specific instrument.

This is the only example where both items in a WSC pair diverged from the expected response.

Gender error: German speakers are often uncertain about the grammatical gender of loanwords or less familiar nouns.

(7) 3 Autos konnten in der Garage parken, aber nur 2 im Carport, da ___ kleiner war.
majority vote: *es* expected response: *er*
3 cars could park in the garage, but only 2 in the carport, because ___ was smaller.

Note that *Carport* is masculine, though even Wiktionary once mistakenly listed it as neuter.[6]

Complexity: Items can be complex due to too many potential antecedents.

(8) Er konnte das Lenkrad in seinem Auto nicht vom Sitz aus erreichen, weil ___ zu niedrig war.
majority vote *es* expected response *er*
He couldn't reach the steering wheel in his car from his seat because ___ was too low.

[6] https://de.wiktionary.org/wiki/Diskussion:Carport

Item	E/M	Error
Die Frau kaufte eine Muschel, um sie ins Aquarium zu stellen, weil ___ schlicht aussah.	es/sie	persp.
Clara beschloss, Gemüse im Ofen anstatt in der Mikrowelle zu kochen, weil ___ das Gemüse knuspriger schmecken ließ.	er/es	sit.
Clara beschloss, Gemüse im Ofen anstatt in der Mikrowelle zu kochen, weil ___ das Gemüse saftiger schmecken ließ.	sie/es	sit.
Es war eine Herausforderung, den Kochtopf im Spülbecken zu waschen, da ___ flach war.	es/er	?
James ging in der Kälte mit einer Jacke anstelle eines Mantels zum Vorstellungsgespräch, weil ___ professionell aussah.	sie/es	sit.
Der Autor wollte den Monolog in der Geschichte verwenden, aber ___ war zu kurz.	sie/er	persp.
Ihre Beziehung verschlechterte sich auf dem Land, frischte jedoch in der Stadt auf, da ___ für sie eine so belebende Atmosphäre war.	sie/es	sit.
Ron wollte das Hühnerfleisch mit einer Gabel anstelle eines Messers zerkleinern, weil ___ besser funktionieren würde.	sie/es	sit.
Sie ging zum Strand und schwamm im Wasser, weil es so ein sonniger Tag war und ___ heiß war.	er/es	sit.
Eva stellte fest, dass die Pflanzen im Gewächshaus durch den Frost gediehen, während die im Garten starben, weil ___ kälter war.	er/es	sit.
Ich fühlte mich wohler, als ich meinen Freund im Haus küsste als im Park, weil ___ ein öffentlicher Ort war.	er/es	sit.
Er konnte das Lenkrad in seinem Auto nicht vom Sitz aus erreichen, weil ___ zu niedrig war.	er/es	compl.
3 Autos konnten in der Garage parken, aber nur 2 im Carport, da ___ kleiner war.	er/es	gender

Table 4: Items with diverging majority vote and expected response (E/M), including error classification (sit.: situational reference, persp.: perspective shift, gender: gender error, compl.: complexity, ?: unclear error source).

The item contains not just two, but four possible antecedents, namely *he*, *steering wheel*, *car*, and *seat*, for two of them it is plausible to be too 'low' in the context (he and seat), and only three are possible by the selectional restrictions of 'niedrig' (low), namely car, seat and steering wheel.

Table 4 summarizes all 13 cases. Notably, all problematic items come from the Wino-X dataset, not our adapted WSC273 items. This may be due to the fact that WSC273 problems were carefully crafted and reviewed by experts, while Wino-X items stem from crowdsourced WinoGrande problems and may lack this level of precision.

C Additional Tables and Graphs

Model	top1	top10	pron
`XLM-RoBERTa`	0.640	0.650	0.640
`GBERT-large`	0.620	0.650	0.640
`BERT-base-german-cased`	0.580	0.590	0.600

Table 5: Accuracy of each model configuration in predicting the human majority vote.

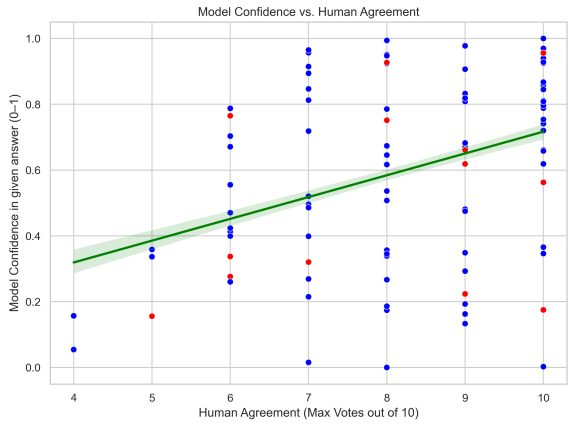


Figure 2: Model confidence (`pron` configuration) as a function of human agreement (maximum number of votes for a pronoun out of 10). Items where the majority vote is incorrect are shown in red.

Finding Answers to Questions: Bridging between Type-based and Computational Neuroscience Approaches

Staffan Larsson[1], Jonathan Ginzburg[2], Robin Cooper[1], Andy Lücking[3]

[1]University of Gothenburg, [2]Université Paris Cité, [3]Goethe-Universität Frankfurt

Abstract

The paper outlines an account of how the brain might process questions and answers in linguistic interaction, focusing on accessing answers in memory and combining questions and answers into propositions. To enable this, we provide an approximation of the lambda calculus implemented in the Semantic Pointer Architecture (SPA), a neural implementation of a Vector Symbolic Architecture. The account builds a bridge between the type-based accounts of propositions in memory (as in the treatments of belief by Ranta, 1994 and Cooper, 2023) and the suggestion for question answering made by Eliasmith (2013), where question answering is described in terms of transformations of structured representations in memory providing an answer. We will take such representations to correspond to beliefs of the agent. On Cooper's analysis, beliefs are considered to be types which have a record structure closely related to the structure which Eliasmith codes in vector representations (Larsson et al., 2023). Thus the act of answering a question can be seen to have a neural base in a vector transformation translatable in Eliasmith's system to activity of spiking neurons and to correspond to using an item in memory (a belief) to provide an answer to the question.

1 Introduction

Understanding how semantic representations are instantiated in biological neural networks remains a fundamental challenge in cognitive science. The Semantic Pointer Architecture (SPA) has been used to build what is currently the world's largest functional brain model (Spaun; Eliasmith, 2013; Eliasmith et al., 2012; Voelker and Eliasmith, 2023), which includes perception, decision making, and motor control systems integrated in a cognitive model that is implemented in spiking neurons and captures detailed anatomical and physiological characteristics of the mammalian brain. The SPA's structured representations are a neural implementation of a Vector Symbolic Architecture (VSA; Gayler, 2003; Schlegel et al., 2022). The basic strategy of the SPA, as we explain below, is to combine a VSA's algebraic structure on a vector space, with coding and decoding operations into ensembles of neurons. In this way, one can retain compositional analyses of natural language in a transparent way that contrasts with LLM approaches, while retaining the robustness and the continuity vectors provide in semantic space. In this paper, we begin to address this challenge by focussing on the processing of questions and answers within a VSA approach, using a VSA approximation of the lambda calculus, and how it can be imlemented in neural network simulations.

Plate (2003, §3.4) and Eliasmith (2013, §4.4) discuss how structured representations encoded as vectors can be manipulated to support reasoning. In particular, on pp. 135ff Eliasmith gives a simple suggestion of how some aspects of question answering could work. In terms of vectors it involves a convolution of a proposition expressed as superpositions of role–filler pairs with another vector corresponding to the question in order to obtain a vector which approximately encodes the answer. However, as Eliasmith (p. 139 2013) himself notes, this model "does not have a solid linguistic justification". Accordingly, VSA approaches to date lack a semantically motivated representation of question and question answering (QA). In particular, some aspects of question and answer processing are still missing, specifically (1) how a proposition containing an answer to a question can be found in memory and (2) how a question and an (elliptical) answer can be combined into a proposition upon hearing the answer. Here we attempt to fill this gap by bridging between a type-based semantic theory of questions and the computational neuroscience VSA approach of the SPA.

The memory which is being probed for answers

Proceedings of the 16th International Conference on Computational Semantics, pages 128–136
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

can be thought of as a collection of proposition encodings – in some cases long term memory (e.g., (1a)), in others working memory, at times a combination thereof (e.g., (1b)).

(1) a. What is the capital of Togo?
b. Are you aware of the wasp on your nose?

Of course, we may not have precisely the proposition we need in memory but may need to reason from a proposition we have to the proposition we need to answer the question.

In this paper we will first explain the formal and conceptual backgrounds that are synthesized in this paper (section 2), including SPA and Type Theory with Records, the semantic framework whose entities underpin our discussion here. We will then explain the previous work on question answering in terms of flat role–filler structures (section 3). Finally, we propose a more comprehensive account of questions and answers using our type-based approach, where we use a SPA approximation of the lambda calculus to handle both question answering and semantic ellipsis resolution (section 4). Section 5 presents a toy model implementation that illustrates and evaluates some features of the SPA-TTR hybrid approach to QA. We conclude in section 6.

2 Background

In this section we describe vector symbolic architectures that mediate between symbolic and distributive representations, neural networks that implement such representations, and finally type-theoretic semantics, specifically the framework Type Theory with Records. In the following, we briefly discuss each of these backgrounds.

2.1 Holographic Reduced Representations (HRR)

Holographic Reduced Representations (HRR; Plate, 2003) are a particular implementation of compressed representations, that is, (higher-order) semantic representations that are obtained by "compressing" (lower-level) semantic representations (Hinton, 1990). HRR achieve this by *circular convolution*: a multiplication operation that binds high-dimensional vectors of dimension d into a new vector of dimension d. Thus, HRR is a true-to-dimension instance of a *Vector Symbolic Architecture* (VSA; Gayler, 2003), in contrast to, for instance, tensor products (Smolensky, 1990).

The vector algebra of HRR includes the following operators ($\mathbf{a}$, $\mathbf{b}$, ... are vectors, i.e., lists of numbers of length d, the dimension of the vector; in the following we usually assume normalized unit vectors, i.e. vectors whose length is 1)[1]:

- $+$: $\mathbf{a}+\mathbf{b} = [\mathbf{a}_0+\mathbf{b}_0, \mathbf{a}_1+\mathbf{b}_1, \ldots, \mathbf{a}_{d-1}+\mathbf{b}_{d-1}]$
- $-$: $\mathbf{a}-\mathbf{b} = [\mathbf{a}_0-\mathbf{b}_0, \mathbf{a}_1-\mathbf{b}_1, \ldots, \mathbf{a}_{d-1}-\mathbf{b}_{d-1}]$
- $\circledast$: $\mathbf{c} = \mathbf{a} \circledast \mathbf{b} : \mathbf{c}_j = \sum_{k=0}^{d-1} \mathbf{a}_k \mathbf{b}_{j-k} \pmod d$
- inverse: $\mathbf{a}' = [\mathbf{a}_0, \mathbf{a}_{d-1}, \mathbf{a}_{d-2}, \mathbf{a}_{d-3}, \ldots, \mathbf{a}_1]$

Basic properties of HRR vector manipulations (Plate, 2003): Circular convolution can be regarded as a multiplication operator for vectors. It has many properties in common with both scalar and matrix multiplication. It is commutative, associative, and bilinear. There is an identity vector and a zero vector and each vector has an approximate inverse ('involution'). Involution distributes over addition and convolution, and is its own inverse. It is noteworthy that the true-to-dimension HRR vector manipualtions are lossy: in particular the inverse $\mathbf{a}'$ of a vector $\mathbf{a}$ is not the exact inverse but approximates it. This "lossiness" introduces the need for clean-up memories when using it in cognitive VSA architectures like the SPA (see below).

2.2 Neural Engineering Framework (NEF)

The *Neural Engineering Framework* (NEF; Eliasmith and Anderson, 2003) implements vectorial representations and manipulations in neural simulations.[2] The basic idea is that high-dimensional vectors figure as the currents that are processed (encoded and decoded) by ensembles of neurons in real time.

2.3 Semantic Pointer Architecture (SPA)

Semantic pointers (Eliasmith, 2013) are structured representations (they can be "dereferenced" to access the more extensive information folded into them) in a (high-dimensional) vector space that function as symbols in cognitive processing and are processed as activity patterns in neural networks. In implementational terms, a semantic pointer can be conceived as a "vector with a name" (it can be addressed); the collection of semantic pointers in this sense make up a "dictionary". In the *Semantic Pointer Architecture* (SPA; Eliasmith, 2013)

[1] The Euclidian length, $\|\cdot\|$, of a vector is the square root of the sum of the squares of its dimension: $\|\mathbf{a}\| = \sqrt{\mathbf{a}_0^2 + \ldots + \mathbf{a}_{d-1}^2}$.

[2] See https://www.nengo.ai/.

of cognitive functions, questions and answers are modeled as semantic pointers (see section 3). SPA uses HRR as default operation for binding (denoted by "⊛") and unbinding (i.e., binding with the inverse vector, below notated with a prime) vectors (see section 2.1), though other algebras can be used in the SPA as well.

Since HRR is lossy (see above), processing with circular convolution "degrade[s] gracefully in the presence of noise" (Plate, 2003, p. 141). To distinguish random noise from "allowable representations", the high-dimensional vectors that are obtained from vector manipulations (carried out by ensembles of neurons in NEF, see section 2.2) are compared to "valid representations" from the vocabularies of semantic pointers (Eliasmith, 2013, §4.6). This validation is derived from inclusion in clean-up memory: the noisy processed vector is validated against the semantic pointers (now conceived as vectors) in the vocabulary of semantic pointers. Technically this is spelled out as the dot product (the standard measure of vector similarity) of the processed vector and the named vectors in the semantic pointer vocabulary.

Clean-up memories are a natural component of lossy VSAs. While the need for long-term clean-up memories is uncontroversial, it is only used sparingly in biological systems because its maintenance is neurologically costly (Stewart et al., 2011). A clean-up memory call replaces a noisy processing vector with its most similar semantic pointer vector from the semantic pointer vocabulary. We indicate the need (or at least the benefit) and the use of clean-up memory as $\mathbf{Clean}(\cdot)$ (or $\mathbf{Clean}_D(\cdot)$ where D indicates the domain of the cleanup function, i.e. the vocabulary which the cleanup function compares with) in the following.

Although being noisy, HRRs involve, among other things, an associative and commutative vector combinatory operation, which is not necessarily the case with other algebraic systems (e.g., Vector-Derived Transformation Binding (Gosmann and Eliasmith, 2019), which can nevertheless be neurally efficient). Futhermore, the SPA (via NEF) relates symbolic, distributional and neural levels. For this reasons, we formulate our approach in terms of the SPA and its default HRR algebra.

2.4 Type Theory with Records (TTR)

We give a brief sketch of those aspects of TTR which we will use in this paper. For more detailed accounts see Cooper (2023).

$s : T$ represents a judgement that s is of type T. Types may be either *basic* or *complex* (in the sense that they are structured objects which have types or other objects introduced in the theory as components). One basic type that we will use is *Ind*, the type of individuals; another is *Real*, the type of real numbers.

Among the complex types are *ptypes* which are constructed from a predicate and arguments of appropriate types as specified for the predicate. Examples are 'man(a)', 'see(a,b)' where $a, b : \mathit{Ind}$. The objects or *witnesses* of ptypes can be thought of as situations, states or events in the world which instantiate the type. Thus $s : \mathrm{man}(a)$ can be glossed as "s is a situation which shows (or proves) that a is a man".

Another kind of complex type are *record types*. In TTR *records* are modelled as a labelled set consisting of a finite set of fields. Each field is an ordered pair, $\langle \ell, o \rangle$, where ℓ is a *label* (drawn from a countably infinite stock of labels) and o is an object which is a witness of some type. No two fields of a record can contain the same label. Importantly, o can itself be a record.

A *record type* is like a record except that the fields are of the form $\langle \ell, T \rangle$ where ℓ is a label as before and T is a type. The basic intuition is that a record, r is a witness for a record type, T, just in case for each field, $\langle \ell_i, T_i \rangle$, in T there is a field, $\langle \ell_i, o_i \rangle$, in r where $o_i : T_i$. (Note that this allows for the record to have additional fields with labels not included in the fields of the record type.) The types within fields in record types may *depend* on objects which can be found in the record which is being tested as a witness for the record type. We use a graphical display to represent both records and record types where each line represents a field. Example (2) represents the type of records which can be used to model situations where a man runs.

(2)
$$\begin{bmatrix} \mathrm{ref} & : & \mathit{Ind} \\ \mathrm{c_{man}} & : & \mathrm{man(ref)} \\ \mathrm{c_{run}} & : & \mathrm{run(ref)} \end{bmatrix}$$

A record of this type would be of the form

(3)
$$\begin{bmatrix} \mathrm{ref} & = & a \\ \mathrm{c_{man}} & = & s \\ \mathrm{c_{run}} & = & e \\ \dots & & \end{bmatrix}$$

where $a : \mathit{Ind}$, $s : \mathrm{man}(a)$ and $e : \mathrm{run}(a)$. Detailed accounts of questions in TTR can be found in (Ginzburg, 2012; Ginzburg et al., 2014, 2022). These are based on viewing questions as akin to

propositional abstracts. This approach is the basis for the most detailed discussion of the response space of questions (Ginzburg et al., 2022) that we are aware of. Another reason for using TTR is that all complex TTR objects are constructed from labelled sets, which correspond to the representation of structured objects which Eliasmith achieves using superposition and circular convolution (Larsson et al., 2023).

2.5 Mapping TTR onto SPA

First steps towards a hybrid of formal and neural semantics by mapping TTR to the SPA have been taken by Larsson et al. (2023). The basic idea was to relate type judgments (not just types) to neural events. By this means, basic types, perceptual and cache-based judgements, singleton types, record types, meet types and merging of record types, ptypes, and subtyping have been accounted for. However, this previous work had little to say about functions, which we address in the following by example of *Wh*-questions.

3 Previous work on question answering in HRR and SPA

Modelling questions in the NEF-SPA means constructing a (biocognitive inspired) network that models a question-related task. Eliasmith (2013) illustrates such a network with question answering (QA). The idea is that the visual cortex provides statements and questions, both supplied as semantic pointers. Statement pointers (e.g., "**red** ⊛ **circle**") represent "the world" and are sent to a memory population of neurons (working memory). Question pointers pose questions to memory content (e.g., "**red**$'$" ≈ "What is red?"). Basal ganglia monitors input and determines what kind of routing is appropriate to answer the question. The answer is sent to the clean-up memory and the memory item with highest similarity is sent to motor cortex (i.e., the answer is given). For instance, the simple statement "dog_{52} chases cat_{43}" could be represented as a flat role–filler structure as follows:

(4) $\textbf{agent} \circledast \textbf{dog}_{52} + \textbf{frame} \circledast \textbf{chase} + \textbf{theme} \circledast \textbf{cat}_{43}$

A fixed set of semantic role labels provides information about where to find the desired information in semantic pointers and can be used for modeling questions. For example, *Who*-questions address entities (persons, or animate beings in general) associated with a certain role in role–filler representations, where addressing is captured in terms of unbinding.

Accordingly, asking the given statement (4) "Who does dog_{52} chase?" amounts to unbinding (4) with **theme**$'$ and thereby retrieving an answer:

(5) $(4) \circledast \textbf{theme}' \approx \textbf{cat}_{43}$

There are obvious ways to make this QA network more complex. Firstly, more complex models need to employ more semantic roles (e.g., location, time, colour, shape, manner, ...), in particular to deal with embedded clauses.

Secondly, a long-term memory along with the working memory will be used. Routing through basal ganglia will then decide when to look into working or long-term memory to find an answer to a given question.

QA with flat role–filler structures in the SPA according to the pattern outlined above offers two insights:

1. The structures of questions and answers have to match closely: the statement pointer that is enquired by a question pointer needs to be tailored to the question asked. In this sense, the answering statement has to be given/known in advance.
2. QA is *dynamic*: regardless of the knowledge source of the answer (e.g., actual perception or long term memory retrieval), both the question and the answer are rehearsed in working memory for QA.

The first issue seems to be a limiting consequence of using the inverses of "label pointers" within flat role–filler structures as models for questions. This assumption is not shared in linguistic semantics, which instead uses *functions*. Linguistic semantics in turn ignores the dynamicity of knowledge source retrieval and QA rehearsal in working memory. Here we aim to reconcile this gap between neuro-computational and formal semantics.

4 Dealing with questions using Lambda calculus in SPA-TTR

The role-filler approach to simple question answering outlined in section 3 assumes that you have already found the proposition which provides the answer and tells you where to look to find the answer within the proposition. A natural language question has to in addition give you material to find the proposition in memory. Also, since answers

are often semantically underspecified (and syntactically elliptical) and thus need to be understood in the context of a specific question, it must be possible to combine questions and (underspecified) answers into full propositions.

An account of question and answer processing thus needs to address the following:

- How are questions and answers combined to propositional types (ptypes)?
- How can answers to a question be found in memory?

We address these questions in sections 4.1 and 4.5, respectively. Along the way, we also cover typechecking of answer candidates in section 4.2, double abstraction ("Who chases who?") in section 4.3, and as a preparation for section 4.5 we also adapt the role-filler approach to extracting an answer from a question and a ptype, in section 4.4.

We will represent questions as functions. The body of the function tells you what would be an appropriate proposition to find in memory. The abstraction in the function tells you where to look in that proposition.

4.1 Combining question and answer into a ptype

We will start by showing how a question and a semantically underspecified (elliptical) answer can be combined into a TTR ptype. For simplicity, we will assume that referents have been identified, so that we write dog_{52} (a specific dog) instead of "the dog" or "that dog". Suppose we have the following exchange:

(6) Q: "Who does dog_{52} chase?" A:"cat_{43}"

In TTR, this would be handled by applying a question q to an answer a to arrive at a ptype p:

(7) a. $q = \lambda v : \mathit{Ind}.\text{chase}(\text{dog}_{52}, v)$
b. $a = \text{cat}_{43}$
c. $p = q(a) = \text{chase}(\text{dog}_{52}, \text{cat}_{43})$

To convert this into SPA-TTR, we use the fact that in TTR, $\lambda v : \mathit{Ind}.\text{chase}(\text{dog}_{52}, v)$ is the labelled set

(8) $$\left\{ \begin{array}{l} \langle \text{lambda}, \mathit{Ind} \rangle, \\ \langle \text{body}, \left\{ \begin{array}{l} \langle \text{pred}, \text{chase} \rangle \\ \langle \text{arg1}, \text{dog}_{52} \rangle \\ \langle \text{arg2}, \text{`body.arg2'} \rangle \end{array} \right\} \rangle \end{array} \right\}$$

In SPA-TTR, we modify this slightly. Firstly, we let the value of the abstracted field be **I**, the identity vector. Secondly, we add a field **lambdapath** containing a path in the question body leading to the abstracted field.

(9) $$\mathbf{Q} = \begin{pmatrix} \mathbf{lambdapath} \circledast \mathbf{arg2}+ \\ \mathbf{lambdatype} \circledast \mathbf{Ind}+ \\ \mathbf{body} \circledast \begin{pmatrix} \mathbf{pred} \circledast \mathbf{chase}+ \\ \mathbf{arg1} \circledast \mathbf{dog}_{52}+ \\ \mathbf{arg2} \circledast \mathbf{I} \end{pmatrix} \end{pmatrix}$$

with **I** the identity vector $\mathbf{I} = [1, 0, 0, \ldots]$, such that for any vector **x**, $\mathbf{I} \circledast \mathbf{x} = \mathbf{x}$. This is a variant of de Bruijn indexing (de Bruijn, 1972), coding lambda terms without using variables but using paths to mark positions instead. (For a different variant using paths see Cooper, 2023.)

For an answer like the one in (10), we want to get the ptype in (11).

(10) $\mathbf{A} = \mathbf{cat}_{43}$

(11) $$\mathbf{P} = \begin{pmatrix} \mathbf{pred} \circledast \mathbf{chase}+ \\ \mathbf{arg1} \circledast \mathbf{dog}_{52}+ \\ \mathbf{arg2} \circledast \mathbf{cat}_{43} \end{pmatrix}$$

In SPA-TTR, **Q** and **A** are inputs to a network for lambda function application that combines them into a proposition by realizing the SPA function below:[3]

(12) $f(Q,A) = Q \circledast \mathbf{body}' - \mathit{Path} + \mathit{Path} \circledast A$
where $\mathit{Path} = Q \circledast \mathbf{lambdapath}'$

For our example above, this gets us

(13) $\mathbf{P} \approx f(\mathbf{Q}, \mathbf{A}) = \mathbf{Q} \circledast \mathbf{body}' - \mathbf{arg2} + \mathbf{arg2} \circledast \mathbf{A}$

This function outputs the body of the question with the lambda abstracted variable replaced by the argument.

For the specific $Q = \mathbf{Q}$ and $A = \mathbf{A}$ given above, $\mathit{Path} = Q \circledast \mathbf{lambdapath}'$ evaluates (with some noise) to **arg2**, yielding

(14) $$\mathbf{P} = \begin{pmatrix} \mathbf{pred} \circledast \mathbf{chase}+ \\ \mathbf{arg1} \circledast \mathbf{dog}_{52}+ \\ \mathbf{arg2} \circledast \mathbf{I} \end{pmatrix} - \mathbf{arg2} + \mathbf{arg2} \circledast \mathbf{cat}_{43} \approx \begin{pmatrix} \mathbf{pred} \circledast \mathbf{chase}+ \\ \mathbf{arg1} \circledast \mathbf{dog}_{52}+ \\ \mathbf{arg2} \circledast \mathbf{cat}_{43} \end{pmatrix}$$

as desired. To reduce noise, we can add a **Clean**() operation over the domain of possible paths:

(15) $\mathit{Path} = \mathbf{Clean}_{\mathbf{Paths}}(Q \circledast \mathbf{lambdapath}')$
where $\mathit{Paths} = \{\mathbf{arg1}, \mathbf{arg2}, \mathbf{pred}\}$

[3] We would like to thank Chris Eliasmith for help with this formulation.

4.2 Typechecking

Above, we have been ignoring the typechecking specified in (7) . One strategy for including it is to prefix the function f in (12) with a term that returns the identity vector **I** if typechecking is successful and noise otherwise. This would mean that f returns noise if typechecking fails, but if it succeeds the result will be identical to using the definition in (12). To achieve this, we can use the fact that the result of binding a vector to its own (approximate) inverse is similar to the identity vector:

(16) $A' \circledast A \approx \mathbf{I}$

The idea is then to use this to compare the type of the answer to the type specified by $\mathbf{Q} \circledast \mathbf{lambdatype}'$. To get the type of the answer, we assume there is a vector **FT** which binds objects to their types, so that that $\mathbf{FT} \circledast A'$ for some object A returns the type of A (assuming for now that each object is of exactly one type in **FT**):

(17) $\mathbf{FT} = \ldots + \mathbf{cat_{45}} \circledast \mathbf{Ind} + \ldots$

The typechecking needed for applying a lambda function Q to an argument A can now be handled by binding with

(18) $(\mathbf{FT} \circledast A') \circledast (Q \circledast \mathbf{lambdatype}')'$

which for (12) gets us

(19) $f(Q,A) =$
$((\mathbf{FT} \circledast A') \circledast (Q \circledast \mathbf{lambdatype}')') \circledast$
$(Q \circledast \mathbf{body}' - Path + Path \circledast A)$

For **Q** and **A** as in (9) and (10), respectivly, we have that $\mathbf{FT} \circledast \mathbf{cat}'_{45} \approx \mathbf{Ind}$ and $\mathbf{Q} \circledast \mathbf{lambdatype}' \approx \mathbf{Ind}$, so that (18) evaluates to

(20) $\mathbf{Ind} \circledast \mathbf{Ind}' \approx \mathbf{I}$

Since operations such as (18) and (19) introduce a lot of noise, it might be necessary to support them with clean-up steps. For instance, in a simple implementation (see section 5) the similarity of (20) drops to 0.39. If the two unbinding sub-steps involved in (18) are cleaned-up in a memory of basic types first, it reaches 1. In both cases, however, the type *Ind* is the most similar base type for the unbound types of question and answer. If we assume that abstracted arguments are of basic types[4] and take **BType** to be the SPA-TTR implementation of TTR basic types, we thus replace (18) with (21).

(21) $\mathbf{Clean}_{\mathbf{BType}}(\mathbf{FT} \circledast A') \circledast$
$\mathbf{Clean}_{\mathbf{BType}}(Q \circledast \mathbf{lambdatype}')'$

This provides a general method for doing typechecking for SPA-TTR lambda functions. For brevity, we will exclude typechecking below.

4.3 Double abstraction

What about questions with double abstraction, such as "Who chased who?"? In TTR, this is done as a double lambda term $\lambda v_1, v_2 : \langle Ind, Ind \rangle .\text{chase}(v_1, v_2)$. We can construct an f_{sim2} (along the lines of the definition of f above) to work directly on that:

(22) $\mathbf{Q2} = \begin{pmatrix} \mathbf{lambdapath1} \circledast \mathbf{arg1} + \\ \mathbf{lambdatype1} \circledast \mathbf{Ind} + \\ \mathbf{lambdapath2} \circledast \mathbf{arg2} + \\ \mathbf{lambdatype2} \circledast \mathbf{Ind} + \\ \mathbf{body} \circledast \begin{pmatrix} \mathbf{pred} \circledast \mathbf{chase} + \\ \mathbf{arg1} \circledast \mathbf{I} + \\ \mathbf{arg2} \circledast \mathbf{I} \end{pmatrix} \end{pmatrix}$

(23) $f_{sim2}(Q,A_1,A_2) = Q \circledast \mathbf{body}' - Path_1 +$
$Path_1 \circledast A_1 - Path_2 + Path_2 \circledast A_2$
where $Path_1 = Q \circledast \mathbf{lambdapath1}'$,
$Path_2 = Q \circledast \mathbf{lambdapath2}'$

As a side note, it is also possible to abstract over the same variable more than once, as in "Who chases herself?", in TTR $\lambda v : Ind.\text{chase}(v, v)$, as shown in (24).

(24) $\mathbf{Q3} = \begin{pmatrix} \mathbf{lambdapath} \circledast \mathbf{(arg1+arg2)} + \\ \mathbf{body} \circledast \begin{pmatrix} \mathbf{pred} \circledast \mathbf{chase} + \\ \mathbf{arg1} \circledast \mathbf{1} + \\ \mathbf{arg2} \circledast \mathbf{1} \end{pmatrix} \end{pmatrix}$

Using our original function f in (12) to apply this question to a single argument yields an instantiated ptype as desired:

(25) $f(\mathbf{Q3}, \mathbf{cat_{45}}) =$

$\begin{pmatrix} \mathbf{pred} \circledast \mathbf{chase} + \\ \mathbf{arg1} \circledast \mathbf{1} + \\ \mathbf{arg2} \circledast \mathbf{1} \end{pmatrix} - (\mathbf{arg1} + \mathbf{arg2}) +$
$(\mathbf{arg1} + \mathbf{arg2}) \circledast \mathbf{cat_{45}} =$

$\mathbf{pred} \circledast \mathbf{chase} + \mathbf{arg1} \circledast \mathbf{cat_{45}} + \mathbf{arg2} \circledast$
$\mathbf{cat_{45}} =$

$\begin{pmatrix} \mathbf{pred} \circledast \mathbf{chase} + \\ \mathbf{arg1} \circledast \mathbf{cat}_{45} + \\ \mathbf{arg2} \circledast \mathbf{cat}_{45} \end{pmatrix}$

In a general account of functions and function application, one would also like to include recursive

[4] This assumption is not generally true, e.g. for "why"-questions. We leave such cases for future work.

function application. In SPA-TTR, a recursively applicable function would correspond to a network that can be applied twice, once for each argument. We leave the specification of recursive function application for future work.

4.4 Extracting answer from question and ptype

If we have a ptype P that we know contains the answer to a question Q, we can use a slightly modified version of Eliasmith's method outlined in section 3.

First, we note that the representation in example (4) corresponds closely to our current representation of ptypes, except for the names of the labels (**pred** instead of **frame**, **arg1** instead of **agent**, **arg2** instead of **theme**).

However, representing the whole question as **theme** does not help us in combining questions and answers into ptypes, so instead we use the more elaborate question seen above. Here, **lambdapath** in Q is bound to **arg1**, and by unbinding it we end up doing the exact same operation as suggested by Eliasmith.

(26) $f_{qp}(Q,P) = P \circledast (Q \circledast \mathbf{lambdapath}')'$

Using the **Q** and **A** from (9) and (10), this gets us

(27) $\mathbf{A} = f_{qp}(\mathbf{Q},\mathbf{P}) = \mathbf{P} \circledast (\mathbf{Q} \circledast \mathbf{lambdapath}')' = \mathbf{P} \circledast (\mathbf{arg2})' = \mathbf{cat}_{43}$

4.5 Extracting the answer from the question and memory

In the general case of question answering, however, we cannot assume we have found the ptype. The real challenge is then to find the answer to a question in LTM, and if you only represent the question as **theme**$'$, this will not be possible. It needs to also include **chase** and **dog**. Since we also have these in our representation of the question, we can find relevant ptypes in LTM.

We should instead only assume there is a record with many different ptypes, one (or several) of which may contain an answer. Previous work (Cooper et al., 2015) developed a notion of a *judgement history* consisting of a set of Austinian propositions encoding judgements that situations s are of types T, $s : T$. Inspired by this but simplifying matters somewhat, we will here use **M** as a name for a labelled set of ptypes indexed by natural numbers:

(28) $\mathbf{M} = \mathbf{1} \circledast P1 + \mathbf{2} \circledast P2 + \ldots + n \circledast Pn$

where Pi is a SPA ptype. Given this, we can outline a procedure for finding the answer A in **M** to a question Q:

1. $B = Q \circledast \mathbf{body}'$
2. Find a P which is similar to B; since $A+B$ is similar to A and to B and to any bundle $A+C$, then if for (the SPA representation of) some natural number n we have $\mathbf{M} \circledast n' \approx B$ we can conclude that $P \approx \mathbf{M} \circledast n'$ is a subtype of B
3. Let A be the value of $Q \circledast \mathbf{lambdapath}'$ in P, i.e. $A = P \circledast (Q \circledast \mathbf{lambdapath}')'$ (unless it's noise)

As an example, assume $Q = \mathbf{Q}$ as in (10) above and $\mathbf{434} \circledast \mathbf{P}$ is in **M**. Then

(29) a. $B = \begin{pmatrix} \mathbf{pred} \circledast \mathbf{chase} + \\ \mathbf{arg1} \circledast \mathbf{dog}_{52} + \\ \mathbf{arg2} \circledast \mathbf{I} \end{pmatrix}$

b. $F = \mathbf{Q} \circledast \mathbf{lambdapath}' \approx \mathbf{arg2}$

Now, since there is an $n = \mathbf{423}$ for which $\mathbf{M} \circledast n' \approx B$, we conclude that

(30) $\mathbf{P} \approx \mathbf{M} \circledast \mathbf{423}'$
$\mathbf{A} = \mathbf{P} \circledast \mathbf{arg2}' \approx \mathbf{cat}$

However, the above is not quite sufficient since it does not specify a SPA mechanism for searching **M**. To address this, we can use the fact that SPA does not distinguish TTR labels from values, so to find the n we are looking for, we can use B as an approximation of the ptype P we are looking for;

(31) $n = \mathbf{M} \circledast B'$

or more robustly

(32) $n = \mathbf{Clean}_{\mathbf{Nat}}(\mathbf{M} \circledast B')$

so that we can define a lookup function that returns the ptype in **M** which is the most similar to some other ptype L:

(33) $SearchM(L) = \mathbf{M} \circledast (\mathbf{Clean}_{\mathbf{Nat}}(\mathbf{M} \circledast L'))'$

so that we can get an answer A:

(34) $\mathbf{A} = SearchM(B) \circledast \mathbf{arg2}'$

Based on this, we can define a function $f_{qm}(Q)$ returning an answer to a question Q from **M**:

(35) $f_{qm}(Q) \approx SearchM(Q \circledast \mathbf{body}') \circledast (Q \circledast \mathbf{lambdapath}')'$

This solution is sensitive to noise, and applying cleanup over a limited domain will help. Fortunately, our question representation provides a type constraint on the possible answers (the **lambdatype** field) that we can use to specify the cleanup domain:

(36) $f_{qm}(Q) \approx \mathbf{Clean}_{Q \circledast \mathbf{lambdatype}'}(SearchM(Q \circledast \mathbf{body}') \circledast (Q \circledast \mathbf{lambdapath}')')$

Exploring whether this retrieval mechanism works similarly to human associative memory is a topic for future research.

5 A simple proof-of-concept model

To see if λ-abstracted question answering is feasible with neurons, we implemented the key steps from Section 4.5 in Nengo (https://www.nengo.ai/).[5] The "knowledge base" **M** from which an answer is to be found consists of six role-filler semantic pointers (vectors of 128 dimensions) that correspond to the statements P1 = *dog chases cat*, P2 = *dog chases cow*, P3 = *dog sees mouse*, P4 = *mouse sees cow*, P5 = *cat likes mouse*, P6 = *dog likes cow*. The inputs to the network are three questions, posed one after the other: 1. *Who does the cat like*? 2. *Who does the dog chase*? 3. *Who likes the cow*? An answer for 1. can be found in P5 (i.e., *mouse*), an answer for 3. can be found in P6 (i.e., *dog*). 2., however, is ambiguous: possible answers are provided by P1 and P2 (i.e., *cat* or *cow*).

The networks unbinds the body and the lambdapath from the question, substracts the lambdapath, and involutes both to retrieve the fragment answer (i.e., compares the resulting vector to the semantic pointers in clean-up memory). The result is shown in the bottom row in fig. 1 ("Answer without Memory Clean-up") and shows that the model does not perform well: the semantic pointer corresponding to *mouse* is always returned as the answer, which is wrong in all but one case. Apparently, the convolution operations introduce too much noise.

To compensate for the noise, we introduced a memory clean-up step (over the domain **Ind**, here limited to **dog**, **cat**, **cow** and **mouse**) after unbinding the questions' bodies. The vector that is fed into an autoassociative clean-up memory over time is most similar to the propositions shown in the top row of fig. 1 ("Memory Input"). The clean-up step reinforces this input (see "Memory Output"). As a consequence, the network now returns vectors that are indeed most similar to the expected items (i.e., *mouse*, *cow* or *cat*, and *dog*, see "Answer with Memory Clean-up"). Note that the last answer is in close competition with the wrong fragment *mouse*, so a final cleanup may be required.

[5]The model can be obtained from https://github.com/aluecking/QA-SPA-TTR.

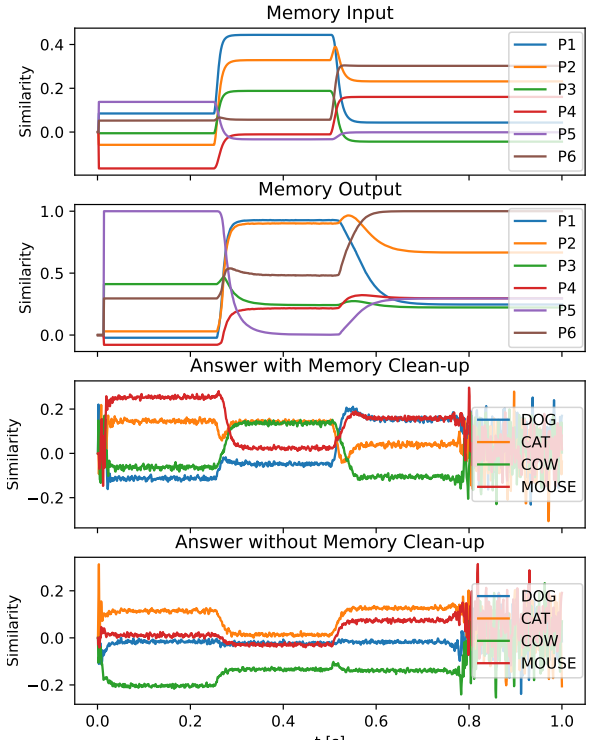


Figure 1: Retrieving a fragment answer according to section 4.5 from a neural simulation. The neural simulation is dynamic because it runs in real time. The elapsed time in seconds is shown on the x-axis. The input to the network changes over time: from 0 s to 0.25 s the input is a semantic pointer that corresponds to the question *Who does the cat like?*, from 0.25 s to 0.5 s to the question *Who does the dog chase?*, and from 0.5 s to 0.72 s to *Who likes the cow?* (no input in the remaining quarter of a second).

6 Conclusions and Future Work

In this paper we have sketched an approach to abstraction, questions, and answering in SPA which allows us to maintain compositional semantic analyses within a biologically plausible cognitive framework. This is of course just the first step for such an account which one should enhance with networks assessing whether a proposition *resolves* a question or constitutes a partial or indirect answer.

Work in VSAs has not to date addressed quantification, but we take this as a first step to showing that this can be done. We hypothesize that the non-Generalized Quantifier TTR-based approach to quantification developed in (Lücking and Ginzburg, 2022) affords a feasible path to this aim. Another important step semantics involves providing an account of working memory, in order to integrate the current insights of dynamic dialogical semantic frameworks such as KoS (Ginzburg, Eliasmith, and Lücking, 2024), MSDRT (Kamp, 2024), and SDRT (Asher and Lascarides, 2003).

References

Nicholas Asher and Alex Lascarides. 2003. *Logics of Conversation*. Cambridge University Press, Cambridge.

Robin Cooper. 2023. *From Perception to Communication: a Theory of Types for Action and Meaning*. Oxford University Press. Open access: https://global.oup.com/academic/product/from-perception-to-communication-9780192871312.

Robin Cooper, Simon Dobnik, Staffan Larsson, and Shalom Lappin. 2015. Probabilistic type theory and natural language semantics. *Linguistic Issues in Language Technology*, 10.

Nicolaas Govert de Bruijn. 1972. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem. *Indagationes Mathematicae (Proceedings)*, 75(5):381–392.

Chris Eliasmith. 2013. *How to build a brain: A neural architecture for biological cognition*. Oxford University Press.

Chris Eliasmith and Charles H. Anderson. 2003. *Neural Engineering*. Computational Neuroscience. MIT Press, Cambridge, MA.

Chris Eliasmith, Terrence C Stewart, Xuan Choo, Trevor Bekolay, Travis DeWolf, Yichuan Tang, and Daniel Rasmussen. 2012. A large-scale model of the functioning brain. *science*, 338(6111):1202–1205.

Ross W. Gayler. 2003. Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience. pages 133–138. Slightly updated 2004 in paper cs/0412059 on arXiv.

Jonathan Ginzburg. 2012. *The Interactive Stance: Meaning for Conversation*. Oxford University Press, Oxford.

Jonathan Ginzburg, Robin Cooper, and Tim Fernando. 2014. Propositions, questions, and adjectives: a rich type theoretic approach. In *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, pages 89–96, Gothenburg, Sweden. Association for Computational Linguistics.

Jonathan Ginzburg, Chris Eliasmith, and Andy Lücking. 2024. Swann's name: towards a dialogical brain semantics. In *Proceedings of the Trentologue, the Twenty Seventh Workshop on the Formal Semantics and Pragmatics of Dialogue*.

Jonathan Ginzburg, Zulipiye Yusupujiang, Chuyuan Li, Kexin Ren, Aleksandra Kucharska, and Paweł Łupkowski. 2022. Characterizing the response space of questions: data and theory. *Dialogue and Discourse*. https://journals.uic.edu/ojs/index.php/dad/article/download/11531/10757.

Jan Gosmann and Chris Eliasmith. 2019. Vector-derived transformation binding: An improved binding operation for deep symbol-like processing in neural networks. *Neural Computation*, 31(5):849–869.

Geoffrey E. Hinton. 1990. Mapping part-whole hierarchies into connectionist networks. *Artificial Intelligence*, 46(1):47–75.

Hans Kamp. 2024. Can't believe it went by so fast. *Annual Review of Linguistics*, 10:1–16.

Staffan Larsson, Robin Cooper, Jonathan Ginzburg, and Andy Lücking. 2023. TTR at the SPA: Relating type-theoretical semantics to neural semantic pointers. In *Proceedings of the 4th Natural Logic Meets Machine Learning Workshop (NALOMA23)*. Association of Computational Linguistics.

Andy Lücking and Jonathan Ginzburg. 2022. Referential transparency as the proper treatment of quantification. *Semantics and Pragmatics*, 15:4.

Tony A. Plate. 2003. *Holographic Reduced Representation*. CSLI Publications, Stanford, CA.

Aarne Ranta. 1994. *Type-Theoretical Grammar*. Clarendon Press, Oxford.

Kenny Schlegel, Peer Neubert, and Peter Protzel. 2022. A comparison of vector symbolic architectures. *Artificial Intelligence Review*, 55(6):4523–4555.

Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial intelligence*, 46(1-2):159–216.

Terrence C. Stewart, Yichuan Tang, and Chris Eliasmith. 2011. A biologically realistic cleanup memory: Autoassociation in spiking neurons. *Cognitive Systems Research*, 12:84–92.

Aaron R Voelker and Chris Eliasmith. 2023. Programming neuromorphics using the neural engineering framework. In *Handbook of Neuroengineering*, pages 1519–1561. Springer.

Can Large Language Models Robustly Perform Natural Language Inference for Japanese Comparatives?

Yosuke Mikami[1,2] **Daiki Matsuoka**[1,2] **Hitomi Yanaka**[1,2]
[1]The University of Tokyo
[2]Riken
{ymikami, daiki.matsuoka, hyanaka}@is.s.u-tokyo.ac.jp

Abstract

Large Language Models (LLMs) perform remarkably well in Natural Language Inference (NLI). However, NLI involving numerical and logical expressions remains challenging. Comparatives are a key linguistic phenomenon related to such inference, but the robustness of LLMs in handling them, especially in languages that are not dominant in the models' training data, such as Japanese, has not been sufficiently explored. To address this gap, we construct a Japanese NLI dataset that focuses on comparatives and evaluate various LLMs in zero-shot and few-shot settings. Our results show that the performance of the models is sensitive to the prompt formats in the zero-shot setting and influenced by the gold labels in the few-shot examples. The LLMs also struggle to handle linguistic phenomena unique to Japanese. Furthermore, we observe that prompts containing logical semantic representations help the models predict the correct labels for inference problems that they struggle to solve even with few-shot examples.

1 Introduction

In recent years, Large Language Models (LLMs) have demonstrated high performance across a wide range of tasks, including Natural Language Inference (NLI; Bowman et al. 2015). However, inference with numerical and logical expressions remains challenging for LLMs (She et al. 2023, Liu et al. 2023a, Parmar et al. 2024). In particular, NLI involving comparatives is important, as it requires a proper understanding of such expressions. Indeed, there are English benchmarks focusing on comparatives for pre-trained models and inference systems (Haruta et al. 2022, Liu et al. 2023b).

However, it has not been thoroughly investigated how *robust* LLMs are in handling various types of inference involving comparatives, regardless of the prompt formats or the few-shot example selection. Moreover, there is growing attention to analyzing the robustness of inference in languages that are not dominant in the pre-training data.

Given these motivations, we construct an NLI dataset focusing on Japanese comparatives by creating templates from an existing Japanese NLI dataset and filling in them with words.[1] Using this dataset, we evaluate five LLMs, including both open and commercial models. We analyze how robustly LLMs can perform inference on comparatives regardless of the way prompts are given in zero-shot and few-shot settings. We also compare LLMs with ccg-jcomp[2] (Mikami et al. 2025), a logical inference system for Japanese comparatives.

The experimental results suggest that the prompt formats impact the model behavior in the zero-shot settings, and that the few-shot performance is influenced by the gold labels in the few-shot examples. In addition, prompts with semantic representations from ccg-jcomp can improve model accuracy on problems that remain difficult even with standard few-shot settings.

2 Related Work

In this section, we describe existing datasets that contain inference problems involving comparatives. JSeM (Kawazoe et al. 2017) is a Japanese NLI dataset, constructed from the English NLI dataset FraCaS (Cooper et al. 1996) with some additional problems that cover inference unique to Japanese. The problems are divided into sections based on semantic phenomena, including comparatives, which allows us to evaluate the strengths and weaknesses of models with respect to individual phenomena. However, since JSeM is limited in vocabulary and small in scale, we create templates from the dataset

[1]Our dataset is available on https://github.com/ynklab/comparativeNLI_dataset
[2]https://github.com/ynklab/ccg-jcomp

Proceedings of the 16th International Conference on Computational Semantics, pages 137–146
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

ID	Category	Template	Example	Label
jsem-570	basic comparative	P: X-wa Y-yori A. X-TOP Y-than A (X is more A than Y)	Taro-wa Hanako-yori omoi. Taro-TOP Hanako-than heavy (Taro is heavier than Hanako)	*unk*
		H: X-wa A. X-TOP A (X is A)	Taro-wa omoi. Taro-TOP heavy (Taro is heavy)	
jsem-577	equative	P: X-wa Y-to onaji-kurai-no N_A-da. X-TOP Y-COM as N_A-COP (X is as A as Y)	Taro-wa Jiro-to onaji-kurai-no omosa-da. Taro-TOP Jiro-COM as weight-COP (Taro is as heavy as Jiro)	*unk*
		H: X-wa Y-yori A. X-TOP Y-than A (X is more A than Y)	Taro-wa Jiro-yori omoi. Taro-TOP Jiro-than heavy (Taro is heavier than Jiro)	
jsem-620	presupposition	P: X-wa Y izyoo-ni A. X-TOP Y than A (X is more A than Y)	Taro-wa Hanako izyoo-ni omoi. Taro-TOP Hanako than heavy (Taro is heavier than Hanako)	*yes*
		H: Y-wa A. Y-TOP A (Y is A)	Hanako-wa omoi. Hanako-TOP heavy (Hanako is heavy)	

Table 1: Examples of categories and their corresponding templates. P and H denote the premise and the hypothesis, respectively. X (Y), A, and N_A are a proper noun, an adjective, and the noun form of an adjective, respectively. ID indicates the ID in the original JSeM dataset. *unk* stands for the *unknown* label.

and generate new problems by filling in the templates with various words.

CAD (Haruta et al. 2022) is a dataset on English adjectives, comparatives, adverbs, and quantifiers. The authors chose inference examples from linguistic papers and constructed new problems by applying transformations such as adding negation and replacing words. Adjective Scale Probe (Liu et al. 2023b) is a dataset designed to investigate how well language models understand degree semantics. It is semi-automatically generated based on templates. While these studies evaluate the extent to which pre-trained language models perform inference involving comparatives in fine-tuned settings, they do not specifically focus on the robustness of the inference in in-context learning settings. To address this gap, we provide a scalable NLI dataset involving Japanese comparatives based on templates created from existing hand-crafted NLI problems.

3 Dataset Creation

To analyze the extent to which LLMs robustly perform inference involving Japanese comparatives, we create an NLI dataset based on the comparatives section of JSeM. Our dataset construction process is composed of (i) template creation based on JSeM and (ii) problem creation using the templates.

3.1 Template Creation

First, for each problem in JSeM, we manually construct a template containing blanks for adjectives, verbs, numerals, and nouns. Each template has at least one premise and one hypothesis. The gold labels are *yes*, *no*, and *unknown*, corresponding to entailment, contradiction, and neutral, respectively.

The templates are classified into ten categories based on JSeM: basic comparative, equative, clausal comparative, numerical, ambiguous, temporal, quantifier, absolute adjective, presupposition, and superlative. One problem may have multiple categories.

Table 1 shows some examples of categories and their corresponding templates. In what follows, we will refer to a template with its original ID in JSeM, which is shown in the leftmost column. First, jsem-570 involves a basic comparative expression *yori*. Second, jsem-577 targets the equative construction, with its premise meaning that the degree of property A is almost the same for X and Y. Since the premise does not specify which degree is greater, its gold label is *unknown*. Third, jsem-620 is one of the problems focusing on the fact that some Japanese comparative expressions trigger a presupposition (Kubota 2012, Hayashishita 2007). Here, the phrase "izyoo-ni" makes the premise presuppose that Y is A, as a result of which the premise entails the hypothesis.

3.2 Problem Creation

We create new problems by filling in the templates with words corresponding to each part of speech, in order to see whether the models can consistently capture the inference patterns independently of specific content words. The words to be inserted into

the templates are carefully chosen by the authors, who are native speakers of Japanese, for their naturalness. In what follows, we detail the concrete procedure for word insertion.

As for a placeholder for an adjective, we insert gradable adjectives in a way that the gold label remains unchanged. More specifically, we avoid using a certain class of adjectives called *absolute adjectives* (Kennedy and McNally 2005), which allow inference from "X is more A than Y" to "Y is A" (e.g., "wet"). Since this property may lead to undesirable changes to the gold label in some templates, we make sure that the inserted word is not an absolute adjective.

In addition, we adopt different strategies depending on whether the placeholder involves the predicative or attributive use. With the predicative use, we insert only adjectives that can take a person as their subject. When the placeholder for an adjective involves the attributive use, in which case the whole template also contains placeholders for a noun and a verb, we construct and apply a list of plausible adjective-noun-verb combinations. More concretely, we first input the template into GPT-4o to generate some adjective-noun-verb combinations. Then, we manually select natural ones from them. To illustrate, consider the template "Taro [verb] a more [adjective] [noun] than Jiro" (for expository purposes, we write the template in English). If the LLM produces the combinations *expensive-car-bought* and *expensive-backpack-drank*, we choose the first output but not the second, since only the first combination results in a semantically-natural sentence when inserted.

Finally, for templates involving numerals, we set a natural range of numerical values compatible with the lexical item for each problem and select the numbers to fill in the templates within that range. For instance, in the template "Taro ate [number] apples," we choose numbers less than 5.

With these strategies, we generate approximately 60 problems from each template. As a result, the total number of problems is 4304, and the distribution of the gold labels is (*yes*/*no*/*unknown*) = (2524/466/1314).

4 Evaluation of Zero-shot NLI

First, we analyze how consistent the performance of the LLMs is regardless of the prompts in the zero-shot prompt setting, compared with a logical inference system.

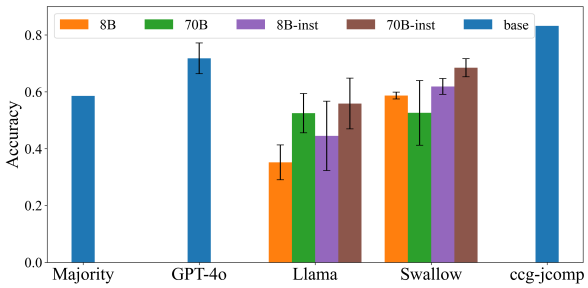


Figure 1: Accuracies on our dataset in the zero-shot setting (average and standard deviation of nine prompts). "Majority" indicates the accuracy achieved by answering *yes*, the most frequent label in the dataset, for all problems.

4.1 Experimental Setting

Models We evaluate five LLMs: GPT-4o[3], Llama-3.1-8B/70B[4] (Llama8B/70B), instruction-tuned Llama-3.1-8B/70B (Grattafiori et al. 2024), Llama-3.1-Swallow-8B/70B[5] (Swallow8B/70B), and instruction-tuned Llama-3.1-Swallow-8B/70B (Fujii et al. 2024). Llama 8B/70B are open-source and multilingual models but do not officially support Japanese. Swallow is a model obtained by performing continual pre-training on Llama with a large Japanese corpus to enhance Japanese language capabilities.

Prompts We conduct experiments using nine different prompts.[6] We create the prompts based on the templates in the FLAN collection (Longpre et al. 2023), which compiles instruction tuning data and methods. The templates contain multiple evaluation instructions, so we use them to examine the models' robustness to prompts. The details of the prompts are shown in Appendix A.

Logical Inference System We also evaluate ccg-jcomp (Mikami et al. 2025), a logical inference system for Japanese comparatives. This system derives semantic representations of the input sentences and performs theorem proving to judge the entailment relation.

4.2 Results and Discussion

Figure 1 presents the accuracy of each system. As shown, GPT-4o demonstrated the best performance

[3] https://openai.com/index/gpt-4o-system-card/
[4] https://huggingface.co/collections/meta-llama/llama-31-669fc079a0c406a149a5738f
[5] https://huggingface.co/collections/tokyotech-llm/llama-31-swallow-66fd4f7da32705cadd1d5bc6
[6] The experiments were conducted in May and June 2025.

of all the LLMs. Among the open-source models, Swallow, which specifically targets Japanese, outperformed Llama. In addition, larger models performed better, and instruction-tuned models outperformed their non-tuned counterparts of the same size. All models had variations depending on the prompt, and these variations were particularly large for Llama8B-inst and Swallow70B.

LLMs tended to produce incorrect answers even for relatively simple problems. For instance, they often incorrectly answered *yes* to the problems generated from jsem-570 in Table 1, possibly due to the lexical overlap between the premise and the hypothesis. Previous studies have suggested that there are lexical overlap heuristics or order-preserving subset heuristics in pre-trained models performing NLI tasks (McCoy et al. 2019, Yanaka and Mineshima 2021). The experimental result indicates that such heuristics may also be present in LLMs.

We also highlight that the LLMs struggled to handle linguistic phenomena that exist in Japanese but not in English. GPT-4o failed to correctly answer the problems related to presupposition (e.g., jsem-620), which is unique to Japanese comparatives. About Llama and Swallow, they tended to incorrectly answer *yes* to problems such as (1), in which (1a) is the premise and (1b) is the hypothesis.

(1) a. Taro-wa Jiro ka Saburo-yori omoi.
Taro-TOP Jiro or Saburo-than heavy
"Taro is heavier than Jiro or Saburo."

b. Taro-wa Jiro-yori omoi.
Taro-TOP Jiro-than heavy
"Taro is heavier than Jiro."

Here, the gold label is *unknown* because the disjunction in (1a) cannot have narrow scope below *than*. In contrast, its English counterpart does allow such a reading (i.e., Taro is heavier than both Jiro and Saburo), making the label *yes*. It is possible that the errors of the models are due to this difference between the two languages.

5 Evaluation of Few-shot NLI

Next, we analyze the extent to which model predictions change depending on how few-shot examples related to the problem category are given.

5.1 Experimental Setting

For GPT-4o, Llama70B-inst, and Swallow70B-inst, we conduct two types of few-shot experiments with the prompt that showed the highest accuracy in Section 4.

Few_normal For each problem, we give the models one few-shot example generated from the same template. For instance, we show an example generated from jsem-570 to a model, and then evaluate it on a modified version where at least one of X, Y, and A is replaced with a different word.

Few_adversarial For each problem, we give the models an example that is closely related to the problem but has a different gold label. For example, when evaluating a model on jsem-577, we give it an example whose premise is augmented with "Y-wa A" (Y is A). This revision changes the gold label to *yes*. Note that we conduct this experiment only for categories with more than one kind of gold label.

5.2 Results and Discussion

Figure 2 shows the accuracies of the three models in each setting. In FEW_NORMAL, all the models showed improved accuracy compared to the zero-shot setting. In particular, Swallow70B-inst exhibited a significantly larger improvement than the other two. In FEW_ADVERSARIAL, the accuracy of GPT-4o showed a slight improvement, whereas Llama70B-inst and Swallow70B-inst exhibited performance degradation, which was especially notable in Swallow70B-inst.

The results of the two experiments indicate that Swallow70B-inst is highly susceptible to the gold labels of few-shot examples. The other two models effectively leveraged the few-shot examples with the same label, and also were not greatly affected when given examples with a different label.

Although the models avoided many of the errors in the zero-shot experiment with the prompts in FEW_NORMAL, the accuracy did not improve sufficiently in some cases. For example, GPT-4o still failed to correctly answer the problems that require an understanding of presuppositions. In addition, the accuracy of Llama70B-inst for the problems such as (1) was zero.

5.3 Analysis with Semantic Representation Prompts

Inspired by Ozeki et al. (2024), we construct few-shot prompts with not only example problems, but also their semantic representations obtained via ccg-jcomp (see Appendix D for details). We instruct LLMs to generate semantic representations of sentences and then infer the entailment label. We conduct experiments on problems with which each model showed low accuracy even with the

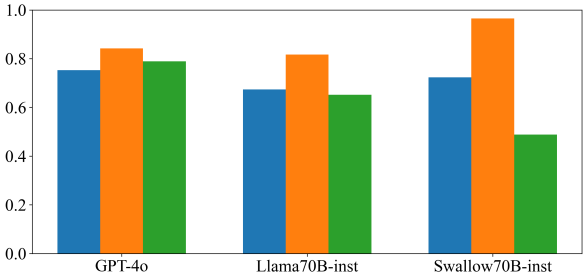


Figure 2: Accuracies of three LLMs in each experimental setting (blue: zero-shot; orange: FEW_NORMAL; green: FEW_ADVERSARIAL)

FEW_NORMAL prompt: namely, presupposition (e.g., jsem-620) for GPT-4o and disjunctive sentences (e.g., (1)) for Llama70B-inst. As a result, the accuracy of GPT-4o and Llama70B-inst increased from 0.049 to 0.230 and from 0.0 to 0.148, respectively. This result suggests that providing semantic representations can improve model performance.

6 Conclusion

In this study, we constructed an NLI dataset focusing on Japanese comparatives, and analyzed how robustly LLMs can perform inference involving comparatives in zero-shot and few-shot settings. The zero-shot experiment revealed that the models' performance varies depending on the prompts, and each model exhibited a distinctive pattern of errors. In the few-shot experiments, we observed that some models, such as Swallow70B-inst, showed a decrease in accuracy when given adversarially designed examples. This observation suggests that some models may be overly sensitive to the specific labels included in the few-shot examples. For problems that the models struggled to solve in the few-shot settings, we found that the accuracy can be improved by making the models predict the semantic representations of the sentences.

Acknowledgments

We thank the three anonymous reviewers for their helpful comments and feedback. This work was partially supported by the Institute for AI and Beyond of the University of Tokyo, and JSPS KAKENHI grant number JP24H00809.

References

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.

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Johan Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, et al. 1996. Using the framework. Technical report, Technical Report LRE 62-051 D-16, The FraCaS Consortium.

Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. 2024. Continual pre-training for cross-lingual llm adaptation: Enhancing japanese language capabilities. *arXiv preprint arXiv:2404.17790*.

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*.

Izumi Haruta, Koji Mineshima, and Daisuke Bekki. 2022. Implementing natural language inference for comparatives. *Journal of Language Modelling*, 10(1):139–191.

J-R Hayashishita. 2007. Izyoo (ni)-and gurai-comparatives: Comparisons of deviation in japanese. *GENGO KENKYU (Journal of the Linguistic Society of Japan)*, 132:77–109.

Ai Kawazoe, Ribeka Tanaka, Koji Mineshima, and Daisuke Bekki. 2017. An inference problem set for evaluating semantic theories and semantic processing systems for japanese. In *New Frontiers in Artificial Intelligence*, pages 58–65, Cham. Springer International Publishing.

Christopher Kennedy and Louise McNally. 2005. Scale structure, degree modification, and the semantics of gradable predicates. *Language*, 81:345 – 381.

Yusuke Kubota. 2012. The presuppositional nature of izyoo (-ni) and gurai comparatives: A note on hayashishita (2007). *GENGO KENKYU (Journal of the Linguistic Society of Japan)*, 141:33–47.

Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. 2023a. Evaluating the logical reasoning ability of chatgpt and gpt-4. *Preprint*, arXiv:2304.03439.

Wei Liu, Ming Xiang, and Nai Ding. 2023b. Adjective scale probe: can language models encode formal semantics information? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13282–13290.

S. Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective

instruction tuning. In *International Conference on Machine Learning*.

Tom 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.

Yosuke Mikami, Daiki Matsuoka, and Hitomi Yanaka. 2025. Implementing a logical inference system for japanese comparatives. In *Proceedings of the 5th Natural Logic Meets Machine Learning Workshop*. Association for Computational Linguistics. To appear.

Kentaro Ozeki, Risako Ando, Takanobu Morishita, Hirohiko Abe, Koji Mineshima, and Mitsuhiro Okada. 2024. Exploring reasoning biases in large language models through syllogism: Insights from the NeuBAROCO dataset. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16063–16077, Bangkok, Thailand. Association for Computational Linguistics.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. 2024. LogicBench: Towards systematic evaluation of logical reasoning ability of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13679–13707, Bangkok, Thailand. Association for Computational Linguistics.

Jingyuan Selena She, Christopher Potts, Samuel R Bowman, and Atticus Geiger. 2023. Scone: Benchmarking negation reasoning in language models with fine-tuning and in-context learning. *arXiv preprint arXiv:2305.19426*.

Hitomi Yanaka and Koji Mineshima. 2021. Assessing the generalization capacity of pre-trained language models through Japanese adversarial natural language inference. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 337–349, Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Prompt Templates

Table 2 shows the prompt templates used in Sections 4 and 5. They are translations of the templates in FLAN related to NLI.

B Results by Category in Zero-shot Experiments

Figure 3 shows the accuracies of each LLM and ccg-jcomp across categories.

C Errors of LLMs in the Zero-shot Experiments

In addition to the errors described in Section 4.2, the LLMs also failed to correctly answer the problems related to equatives such as jsem-577-1 in Table 1. They tended to answer *no*, which suggests that they interpret the premise as meaning that the degrees of the two people are exactly equal.

D Details of the Experiment with Semantic Representation Prompts

Table 3 shows the instruction and a few-shot example used in Section 5.3. It provides the semantic representations adopted in ccg-jcomp.

As for the experimental results, although the accuracy of Llama 70B Instruct was still low compared to other models, the semantic representations it predicted were correct in most problems. Most of the errors stemmed from the reasoning step. Table 4 is an example of reasoning errors. The semantic representations are correct; the model successfully interpreted the premise as "Taro is kinder than Jiro, or Taro is kinder than Saburo." However, it incorrectly concluded that the hypothesis follows the premise.

Template	Translation
{premises} 選択肢付きの質問です：上記の段落に基づいて「{hypothesis}」と結論付けることはできますか。 選択肢：含意、矛盾、中立 回答：	({premises} Question with options: Based on the paragraph above can we conclude that "{hypothesis}"? options: entailment, contradiction, neutral answer:)
{premises} この段落に基づいて、下の文が真であると結論付けることはできますか。 {hypothesis} 選択肢：含意、矛盾、中立 回答：	({premises} Based on that paragraph can we conclude that the sentence below is true? {hypothesis} options: entailment, contradiction, neutral answer:)
{premises} 選択肢付きの質問です：以下の結論を導くことはできますか。 {hypothesis} 選択肢：含意、矛盾、中立 回答：	({premises} Q with options: Can we draw the following conclusion? {hypothesis} options: entailment, contradiction, neutral answer:)
{premises} 前の文が与えられたとき、この次の文は従いますか。 {hypothesis} 選択肢：含意、矛盾、中立 回答：	({premises} Does this next sentence follow, given the preceding text? {hypothesis} options: entailment, contradiction, neutral answer:)
{premises} 選択肢：含意、矛盾、中立 問題：次の文を推論できますか。 {hypothesis} 回答：	({premises} options: entailment, contradiction, neutral Question: Can we infer the following? {hypothesis} answer:)
次の段落を読んで仮説が真かどうかを決定してください。最後の選択肢の中から選んでください： {premises} 仮説：hypothesis 選択肢：含意、矛盾、中立 回答は	(Read the following paragraph and determine if the hypothesis is true. Select from options at the end: {premise} Hypothesis: {hypothesis} options: entailment, contradiction, neutral answer:)
テキストを読んで文が真かどうかを決定してください： {premises} 文：{hypothesis} 選択肢：含意、矛盾、中立 回答：	(Read the text and determine if the sentence is true: {premises} Sentence: {hypothesis} options: entailment, contradiction, neutral answer:)
選択肢付きの質問です：以下の文脈から仮説を導くことはできますか。 文脈： {premises} 仮説：{hypothesis} 選択肢：含意、矛盾、中立 回答：	(Question with options: can we draw the following hypothesis from the context? Context: {premises} Hypothesis: {hypothesis} options: entailment, contradiction, neutral answer:)
次の文が真かどうかをその下のテキストに基づいて決定してください。選択肢から選んでください。 {hypothesis} {premises} 選択肢：含意、矛盾、中立 回答：	(Determine if the sentence is true based on the text below. Choose from options. {hypothesis} {premises} options: entailment, contradiction, neutral answer:)

Table 2: Prompt templates used in Section 4

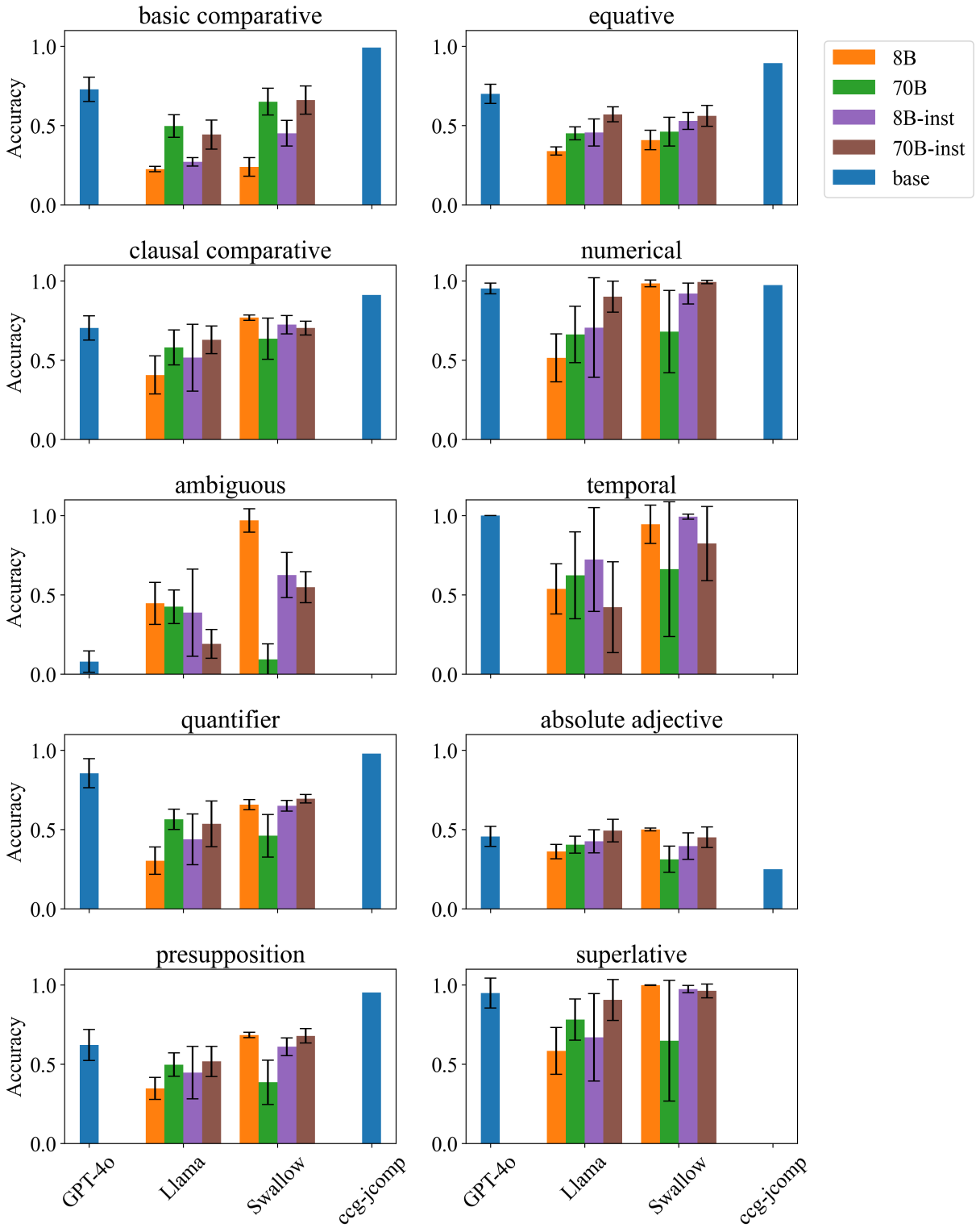


Figure 3: Accuracies of each model and system across categories.

与えられた前提と仮説の間の正しい論理関係を決定してください。
- 仮説が前提から論理的に導かれる場合は「含意」と答えてください。
- 前提と仮説が論理的に両立しない場合は「矛盾」と答えてください。
- 「含意」でも「矛盾」でもない場合は「中立」と答えてください。

入力
前提：太郎は次郎か三郎より明るい。
仮説：太郎は次郎より明るい。

述語論理への翻訳
前提：∃d (明るい(太郎, d) ∧¬明るい(次郎, d)) ∨∃d (明るい(太郎, d) ∧¬明るい(三郎, d))
仮説：∃d (明るい(太郎, d) ∧¬明るい(次郎, d))

推論
[その答えに対する理由を説明してください]

回答
[含意、矛盾、中立のいずれかを答えてください]

(Determine the correct logical relationship between the given premises and the hypothesis.
- Answer “entailment” if the hypothesis follows logically from the premises.
- Answer “contradiction” if the premises and the hypothesis are logically incompatible with each other.
- Answer “neutral” if the relationship is neither “entailment” nor “contradiction.”

Input
Premise: Taro is more cheerful than Jiro or Saburo.
Hypothesis: Taro is more cheerful than Jiro.

Translation into predicate logic
Premise: ∃d (cheerful(taro, d) ∧¬cheerful(jiro, d)) ∨∃d (cheerful(taro, d) ∧¬cheerful(saburo, d))
Hypothesis: ∃d (cheerful(taro, d) ∧¬cheerful(jiro, d))

Reasoning
[Explain your reasoning for the answer]

Answer
[Answer entailment, contradiction, or neutral])

Table 3: Instruction and a few-shot example used in the experiment in Section 5.3. The lower part is an English translation.

Input	前提：太郎は次郎か三郎より優しい。 仮説：太郎は次郎より優しい。 (Premise: Taro is kinder than Jiro or Saburo. Hypothesis: Taro is kinder than Jiro.)
Semantic Representations	前提：∃d(優しい(太郎, d) ∧¬優しい(次郎, d)) ∨∃d(優しい(太郎, d) ∧¬優しい(三郎, d)) 仮説：∃d(優しい(太郎, d) ∧¬優しい(次郎, d)) (Premise: ∃d(kind(taro, d) ∧¬kind(jiro, d)) ∨∃d(kind(taro, d) ∧¬kind(saburo, d)) Hypothesis: ∃d(kind(taro, d) ∧¬kind(jiro, d)))
Reasoning	前提が真であれば、太郎は次郎より優しい、または太郎は三郎より優しい。仮説が真であれば、太郎は次郎より優しい。したがって、仮説は前提から論理的に導かれる。 (If the premise is true, then Taro is kinder than Jiro, or Taro is kinder than Saburo. If the hypothesis is true, then Taro is kinder than Jiro. Therefore, the hypothesis is logically derived from the premise.)

Table 4: Example of reasoning errors of Llama70B-inst. Semantic representations and reasoning are the output.

Is neural semantic parsing good at ellipsis resolution, or isn't it?

Xiao Zhang
Center for Language and Cognition
University of Groningen
xiao.zhang@rug.nl

Johan Bos
Center for Language and Cognition
University of Groningen
johan.bos@rug.nl

Abstract

Neural semantic parsers have shown good overall performance for a variety of linguistic phenomena, reaching semantic matching scores of more than 90%. But how do such parsers perform on strongly context-sensitive phenomena, where large pieces of semantic information need to be duplicated to form a meaningful semantic representation? A case in point is English verb phrase ellipsis, a construct where entire verb phrases can be abbreviated by a single auxiliary verb. Are the otherwise known as powerful semantic parsers able to deal with ellipsis or aren't they? We constructed a corpus of 120 cases of ellipsis with their fully resolved meaning representation and used this as a challenge set for a large battery of neural semantic parsers. Although these parsers performed very well on the standard test set, they failed in the instances with ellipsis. Data augmentation helped improve the parsing results. The reason for the difficulty of parsing elided phrases is not that copying semantic material is hard, but that they usually occur in linguistically complicated contexts, causing most of the parsing errors.

1 Introduction

Semantic parsing is the task of providing a formal meaning representation for an input sentence of a natural language such as English, Dutch, or Italian. Semantic parsing is crucial for applications that require the precise translation of unstructured data (i.e., text and images) into structured data (e.g., databases and robot commands). Currently, the most promising approaches to semantic parsing are based on neural models (Bai et al., 2022; Wang et al., 2023; Zhang et al., 2024b, 2025) trained or fine-tuned on large semantically annotated corpora (Banarescu et al., 2013; Abzianidze et al., 2017), reaching high performance with F scores greater than 90%. Little is known about the ability of neural semantic parsers to cope with *ellipsis*, a linguistic construction in which elements are omitted and are supplied by the discourse context. In this paper, we will study how neural semantic parsers deal with Verb Phrase Ellipsis (VPE) in English. An example of a VPE is shown in (1) together with its fully expressed surface interpretation in (2).

(1) Ann likes grapes, and Bea does, too.
(2) Ann likes grapes, and Bea does **like grapes**, too.

As this very simple example already demonstrates, ellipsis interpretation is a challenging task, for the only way to recover the elided material is to consider the discourse context. The (computational) linguistics literature abounds with many more complicated examples of VPE, including sloppy-strict interpretation of pronouns appearing in the elided material, cascaded ellipsis, antecedent contained deletion, gapping, and embedded ellipsis (Dahl, 1973; Williams, 1977; Roberts, 1989; Dalrymple et al., 1991). Nevertheless, our aim is not to focus on these linguistically interesting examples carefully crafted by linguists, but rather to investigate how data-driven semantic parsers deal with instances of VPE found in corpora.

As far as we know, this is the first in-depth study of VPE interpretation in neural semantic parsing. Related, but taking a different perspective, is work by Hardt (2023), who found that large language models have difficulty processing ellipsis.

In Section 2 we give an overview of earlier computational approaches to VPE. In Section 3 we introduce the Parallel Meaning Bank (PMB) and a VPE challenge test set distilled from the PMB. In Section 4 we outline our approach to enhance the semantic parsing for VPE, while in Section 5 the parsing results are presented, showing that neural approaches face a difficult time in interpreting elliptical constructions, even with substantial fine-tuning, but not for the reasons we initially thought would cause the difficulty.

Proceedings of the 16th International Conference on Computational Semantics, pages 147–152
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

2 Background

VPE interpretation has drawn considerable attention in formal linguistics (Dahl, 1973; Sag, 1976; Klein, 1987; Dalrymple et al., 1991). These early approaches can be summarized as identifying an antecedent verb phrase in the context, providing a logical form while abstracting over the subject, and applying the result to the subject noun phrase of the elided verb phrase. Computational approaches were introduced later (Alshawi, 1992; Kehler, 1993; Bos, 1994; Crouch, 1995; Hardt, 1997), with the landmark paper by Dalrymple et al. (1991) introducing a set of benchmark VPE examples and a sophisticated algorithm based on higher-order unification to construct fully resolved meaning representations for elliptical phrases. These approaches, although computational of nature, still required external modules to identify the source verb phrase and the parallel elements between source and target phrase.

Data-driven approaches based on annotated corpora (Nielsen, 2005; Bos and Spenader, 2011; Bos, 2016) demonstrated the large gap between theoretical ideas and practical implementations (McShane and Babkin, 2016; Kenyon-Dean et al., 2020; Zhang et al., 2019), and were considered to be specific tasks rather than an integral part of wide-coverage semantic parsing. In this paper, we take a different computational perspective and depart with an overall well-performing general-purpose semantic parsing and investigate how well it succeeds on ellipsis data.

3 Data

The Parallel Meaning Bank The PMB Abzianidze et al., 2017 is a multilingual corpus enriched with semantic annotations, covering a wide range of linguistic phenomena. It contains a substantial set of parallel texts, each paired with a formal meaning representation known as a Discourse Representation Structure (DRS) based on Discourse Representation Theory (DRT, Kamp and Reyle, 1993). While DRSs are typically presented in a human-readable box format, a clause-based linear representation was introduced by van Noord et al. (2018) to enable their use in sequence-based models. More recently, Bos (2023) proposed Sequence Box Notation (SBN), a simplified, variable-free version of DRS aimed at further facilitating sequence processing. In this paper we use SBN as meaning representation format (see Figure 1).

Sentence with VPE:
Life never ends but earthly life does.

DRS:

```
life.n.11                          % Life never   [0-10]
          NEGATION <1              %
end.v.01  Theme -1 Time +1         % ends         [11-15]
time.n.08 EQU now                  %
          CONTRAST <2              %
earthly.a.01 AttributeOf +1        % but earthly  [16-27]
life.n.11                          % life         [28-32]
end.v.01  Theme -1 Time +1         % does.        [33-38]
time.n.08 EQU now                  %
```

DRG:

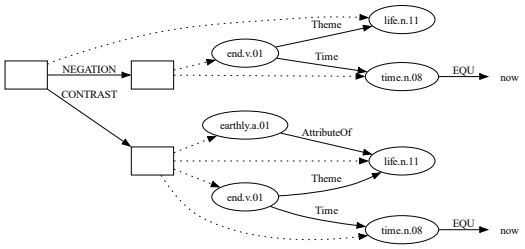


Figure 1: An example sentence with Verb Phrase Ellipsis and meaning representation in sequence notation and drawn as a directed acyclic graph.

Annotated VPE Instances As occurrences of VPE are relatively rare (Bos and Spenader, 2011), it is rather challenging to yield a reasonably sized corpus. A total of 120 cases were identified in the PMB and their corresponding meaning representations manually corrected. Slightly more than half of the cases (71) contained some kind of negation in the elided construction (e.g., ”and neither am I”, ”Greenland is not”, ”but she didn’t”). Half of the instances are accompanied by the auxiliary verb *to do*, a third by *to be*, and the remaining cases are formed by other auxiliary verbs, the infinitival particle *to* or instances of gapping. An annotated example taken from the corpus is shown in Figure 1, where the elliptical phrase ”life does” is semantically interpreted as ”life does end”.

4 Experimental Setup

Training Sets For training our neural semantic parsers, we consider two settings: (1) the **Standard Training Set**, the default training data provided by PMB version 5.1.0, with all texts that are included in the VPE test set removed; and (2) the **Augmented Training Set**, an augmented dataset to enhance the model’s ability to handle verb phrase ellipsis. We construct the augmented dataset apply-

ing the following data augmentation strategies:

- We employ GPT-4 to generate 600 sentence pairs, each consisting of a sentence containing VPE and its corresponding resolved version (i.e., the full sentence with the elided verb phrase explicitly restored in the surface text).
- We use the state-of-the-art DRS parser from Zhang et al. (2024a) to generate DRSs for the resolved sentences. These DRSs are then paired with the original VPE sentences as their target semantic representations.
- We incorporate the generated VPE data into the standard training set in varying quantities (from 100 up to 600) to examine how the scale of augmentation affects model performance and to identify the point at which performance improvements begin to converge.

Test sets We evaluate the trained parsers on two test sets: the **Standard Test Set**, which serves as a general, broad-coverage set for comparison, and **VPE120**, a targeted test set focusing on VPE, as described in Section 3.

Evaluation We evaluate model performance using two metrics: **Smatch**[1] and **Ill-Formed Rate (IFR)**. Smatch (Cai and Knight, 2013; Opitz, 2023) measures the similarity between the predicted and reference semantic graphs by converting each graph into a set of triples and computing the optimal variable mapping via a hill-climbing algorithm. Precision (P), recall (R), and F1 score are calculated as follows:

$$\mathrm{P} = \frac{m}{p}, \quad \mathrm{R} = \frac{m}{g}, \quad \mathrm{F1} = \frac{2 \cdot \mathrm{P} \cdot \mathrm{R}}{\mathrm{P} + \mathrm{R}}, \tag{1}$$

where m denotes the number of matching triples, p is the number of predicted triples, and g is the number of gold-standard triples.

To assess the structural validity of generated graphs, we additionally report the **Ill-Formed Rate (IFR)**. A graph is considered ill-formed if it exhibits structural defects such as cyclic dependencies, isolated nodes, or dangling edges referencing non-existent elements. Graphs identified as ill-formed are assigned a Smatch score and F1 score of zero, thereby contributing to a quantitative measure of structural failure.

[1]We adopt the Smatch++ implementation (Opitz, 2023), which uses Integer Linear Programming (ILP) instead of the standard hill-climbing approach.

Models We evaluated three encoder–decoder models–mBART (Liu et al., 2020), mT5 (Xue et al., 2021), and ByT5 (Xue et al., 2022), as well as four decoder-only models: Qwen2.5-7B (Yang et al., 2024), Ministral-8B, LLaMA3.1-8B (Grattafiori et al., 2024), and Gemma2-9B (Team et al., 2024).

5 Results and Analysis

The performance of models on both test sets is presented in Table 1. Overall, sentences containing VPE instances pose significantly greater challenges for semantic parsing, as evidenced by substantially lower Smatch scores and elevated ill-formed rates (IFR). We analyze these results in detail below.

Table 1: Smatch and IFR performance on the Standard Test Set and VPE120 for models trained with the Standard Training Set, Aug300, and Aug600.

Model	Train set	Standard Test		VPE120	
		Smatch	IFR	Smatch	IFR
mBart-Large	Standard	83.50	6.95	70.90	33.17
	Aug300	85.40	7.00	77.90	27.33
	Aug600	85.00	6.60	78.10	24.83
mT5-Large	Standard	82.61	11.20	70.38	29.83
	Aug300	84.50	9.80	75.20	24.83
	Aug600	84.00	9.20	75.50	24.00
ByT5-Large	Standard	91.40	8.73	66.22	27.33
	Aug300	92.50	7.50	73.00	22.33
	Aug600	92.90	7.00	72.50	22.33
Qwen2.5-7B	Standard	94.19	5.34	77.09	17.33
	Aug300	94.35	5.17	85.31	9.83
	Aug600	95.50	5.09	84.64	12.33
Ministral-8B	Standard	95.45	4.67	82.77	13.17
	Aug300	95.50	4.25	89.00	6.50
	Aug600	95.42	4.59	90.61	6.50
LLaMA3.1-8B	Standard	95.56	4.51	83.11	12.33
	Aug300	95.32	5.18	88.89	12.33
	Aug600	95.44	4.76	89.21	8.17
Gemma2-9B	Standard	96.31	4.59	78.09	17.33
	Aug300	96.46	4.42	88.52	7.33
	Aug600	96.59	4.09	89.20	8.17

Performances on Standard Test Decoder-only architectures consistently outperform encoder–decoder models on the Standard Test set. Gemma2-9B achieves the highest performance with a Smatch score of 96.59 following augmentation (compared to 96.31 on the standard training set). Other decoder-only models demonstrate similarly strong performance: LLaMA3.1-8B (95.44), Ministral-8B (95.50), and Qwen2.5-7B (95.50) all maintain scores above 95. In contrast, encoder–decoder architectures (mBART-Large, mT5-Large, and ByT5-

Large) achieve lower performance, with *standard* scores ranging from 82.61 to 91.40. This performance disparity likely stems from both architectural differences and parameter scale advantages, where larger decoder-only models may benefit from more stable fine-tuning dynamics and in-context learning capabilities.

Performances on VPE120 All models exhibit substantially degraded performance on VPE120 relative to the Standard Test, confirming the inherent difficulty of parsing elliptical constructions semantically. When trained solely on the standard dataset, models achieve VPE120 Smatch scores between 66.22 and 83.11, accompanied by markedly increased IFR (e.g., 33.17% for mBART-Large and 29.83% for mT5-Large), indicating frequent generation of malformed outputs.

VPE-specific data augmentation yields substantial improvements across all architectures. Ministral-8B achieves the highest score of 90.61 with Aug600, closely followed by LLaMA3.1-8B (89.21) and Gemma2-9B (89.20). These top-performing models also demonstrate the most significant IFR reductions (e.g., Ministral-8B: 13.17% $\rightarrow$ 6.50%). Encoder–decoder models also benefit from augmentation: mBART-Large improves from 70.90 (standard) to 78.10 (Aug600), while mT5-Large advances from 70.38 to 75.50. Notably, ByT5-Large shows improvement from 66.22 to 73.00 with Aug300.

These findings demonstrate that VPE-specific data augmentation effectively narrows the performance gap between the Standard and VPE test sets, particularly for larger decoder-only models. The convergence of performance scores beyond Aug300 (see Figure 2) suggests diminishing gains from additional augmentation data, indicating that current models may be approaching their capacity limits for ellipsis resolution. This shows the need for more advanced architectures or specialized training strategies to further improve performance on complex elliptical phenomena.

Qualitative Analysis The previous section showed that sentences with ellipsis are a lot harder to parse for the neural semantic models. But why is this the case? Is this because they are a bad at copying semantic information, or is it something else? In order to answer we examined the output of the best performing model and manually inspected the results.

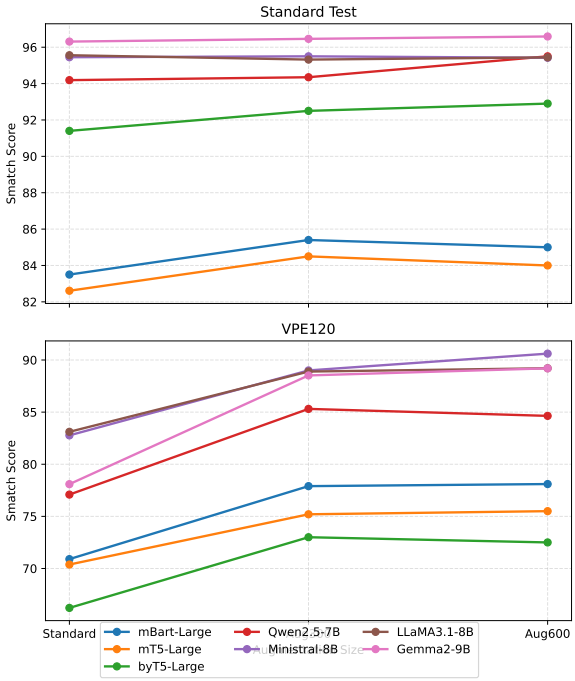


Figure 2: Model performance on VPE120 with increasing augmentation sizes (100 to 600).

Surprisingly, what we thought would be hard for the models, copying semantic material from the source to the target, was not hard at all. Only in three of the 120 cases did this not happen. Actually, what contributed to the low score was the wrong choice of discourse relation (22% of overall errors), the wrong attachment of a discourse relation (20%), incorrect scope order between tense and negation (16%), incorrect choice of word sense (16%), incorrect choice of thematic role (10%), incorrect choice of concept (10%), and incorrectly resolved anaphora (4%).

One reason why selecting the correct VP antecedent might have to do with the amount of ambiguity, or lack thereof. For instance, in the VPE example in Figure 1 there is only one potential verb phrase that could serve as antecedent for the elliptical phrase. Closer inspection of the dataset reveals that most (81%) of the texts with VPE are relatively short and provide only one verb phrase that could act as antecedent; only 23 examples provide two or more potential verb phrase antecedents, as in (3).

(3) Ann hoped to succeed, but she didn't.

Here there are two verb phrases in the context: *hope to succeed* and *succeed*. For most of these cases picking the most recent verb phrase usually yields the correct interpretation.

6 Conclusion

Although open-domain semantic parsing achieves good overall performance on the standard test sets, its shortcomings arise at the surface when looking at more complex linguistic phenomena. We demonstrated this by looking specifically at how neural parsing models deal with cases of English VP Ellipsis. Although we observed a drop in performance, the reason for the drop was not the context-sensitive nature of ellipsis, but rather the fact that elliptical phenomena are often surrounded by complex phenomena such as tense, negation, and discourse structure, causing parsing errors. So, is neural semantic parsing good at ellipsis resolution? Yes, it is!

Acknowledgements

We would like to thank the three anonymous reviewers for their comments. Reviewer 1 pointed out that the developed datasets will be valuable for the community, and indeed the VPE dataset will be made public via the Parallel Meaning Bank data releases. Reviewer 2 wondered how many different VP targets there are for each case of ellipsis, as the complexity for ellipsis resolution "depends greatly on the number of available VP targets for each VPE". We added a discussion in this topic in Section 5. Reviewer 3 noted that the question in the paper's title is not actually answered. This was a correct observation. But now we did explicitly in Section 6. Special thanks go to Juri Opitz, who pointed out that using a hill-climber for evaluation (as we did in the submitted version of this paper) is not optimal. So we recalculated our scores using ILP instead.

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 242–247, Valencia, Spain.

Hiyan Alshawi, editor. 1992. *The Core Language Engine*. The MIT Press, Cambridge, Massachusetts.

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015, Dublin, Ireland. 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.

Johan Bos. 1994. Presupposition & vp ellipsis. In *Proceedings of the 15th International Conference on Computational Linguistics (Coling 1994)*, pages 1184–1190, Kyoto, Japan.

Johan Bos. 2016. Verb phrase ellipsis and sloppy identity: a corpus-based investigation. In Martijn Wieling, Martin Kroon, Gertjan Van Noord, and Gosse Bouma, editors, *From Semantics to Dialectometry*, volume 32 of *Tributes*, pages 57–64. College Publications.

Johan Bos. 2023. The sequence notation: Catching complex meanings in simple graphs. In *Proceedings of the 15th International Conference on Computational Semantics (IWCS 2023)*, pages 1–14, Nancy, France.

Johan Bos and Jennifer Spenader. 2011. An annotated corpus for the analysis of vp ellipsis. *Language Resources and Evaluation*, 45(4):463–494.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Richard Crouch. 1995. Ellipsis and Quantification: A Substitutional Approach. In *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics*, pages 229–236, Dublin, Ireland.

Östen Dahl. 1973. On so-called'sloppy identity'. *Synthese*, pages 81–112.

Mary Dalrymple, Stuart M. Shieber, and Fernando C.N. Pereira. 1991. Ellipsis and Higher-Order Unification. *Linguistics and Philosophy*, 14:399–452.

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*.

Daniel Hardt. 1997. An Empirical Approach to VP Ellipsis. *Computational Linguistics*, 23(4):525–541.

Daniel Hardt. 2023. Ellipsis-dependent reasoning: A new challenge for large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, volume 2, pages

39–47, United States. Association for Computational Linguistics. The 61st Annual Meeting of the Association for Computational Linguistics ; Conference date: 09-07-2023 Through 14-07-2023.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Kluwer, Dordrecht.

Andrew Kehler. 1993. A discourse copying algorithm for ellipsis and anaphora resolution. In *Proceedings of EACL*, pages 203–212.

Kian Kenyon-Dean, Edward Newell, and Jackie Chi Kit Cheung. 2020. Deconstructing word embedding algorithms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8479–8484, Online. Association for Computational Linguistics.

Ewan Klein. 1987. VP Ellipsis in DR Theory. In Jeroen Groenendijk et al., editors, *Studies in Discourse Representation Theory and the Theory of Generalised Quantifiers*, volume 8, pages 161–187. FLORIS, Dordrecht.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Marjorie McShane and Petr Babkin. 2016. Detection and resolution of verb phrase ellipsis. *Linguistic Issues in Language Technology*, 13.

Leif Arda Nielsen. 2005. *A corpus-based study of Verb Phrase Ellipsis Identification and Resolution*. Ph.D. thesis, King's College London.

Rik van Noord, Lasha Abzianidze, Antonio Toral, and Johan Bos. 2018. Exploring neural methods for parsing discourse representation structures. *Transactions of the Association for Computational Linguistics*, 6:619–633.

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.

Craige Roberts. 1989. Modal subordination and pronominal anaphora in discourse. *Linguistics and Philosophy*, 12(6):683–721.

Ivan Sag. 1976. *Deletion and Logical Form*. Ph.D. thesis, MIT.

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*.

Chunliu Wang, Huiyuan Lai, Malvina Nissim, and Johan Bos. 2023. Pre-trained language-meaning models for multilingual parsing and generation. In *Findings of the Association for Computational Linguistics*, page 5586–5600. Association for Computational Linguistics (ACL).

Edwin Williams. 1977. Discourse and logical form. *Linguistic Inquiry*, 8(1):101–139.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

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.

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 e-prints*, pages arXiv–2412.

Wei-Nan Zhang, Yue Zhang, Yuanxing Liu, Donglin Di, and Ting Liu. 2019. A neural network approach to verb phrase ellipsis resolution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7468–7475.

Xiao Zhang, Gosse Bouma, and Johan Bos. 2025. Neural semantic parsing with extremely rich symbolic meaning representations. *Computational Linguistics*, 51(1):235–274.

Xiao Zhang, Qianru Meng, and Johan Bos. 2024a. Retrieval-augmented semantic parsing: Using large language models to improve generalization. *arXiv preprint arXiv:2412.10207*.

Xiao Zhang, Chunliu Wang, Rik van Noord, and Johan Bos. 2024b. Gaining more insight into neural semantic parsing with challenging benchmarks. In *Proceedings of the Fifth International Workshop on Designing Meaning Representations @ LREC-COLING 2024*, pages 162–175, Torino, Italia. ELRA and ICCL.

Extracting Behaviors from German Clinical Interviews in Support of Autism Spectrum Diagnosis

Margareta A. Kulcsar
École Normale Supérieure Paris-Saclay
margareta.kulcsar@ens-paris-saclay.fr

Ian Paul Grant
Queen Mary University London
i.p.grant@qmul.ac.uk

Massimo Poesio
Queen Mary University London
Utrecht University
m.poesio@qmul.ac.uk

Abstract

Accurate identification of behaviors is essential for diagnosing developmental disorders such as Autism Spectrum Disorder (ASD). We frame the extraction of behaviors from text as a specialized form of event extraction grounded in the TimeML framework and evaluate two approaches: a pipeline model and an end-to-end model that directly extracts behavior spans from raw text. We introduce two novel datasets: a new clinical annotation of an existing Reddit corpus of parent-authored posts in English and a clinically annotated corpus of German ASD diagnostic interviews. On the English dataset, the end-to-end BERT model achieved an F1 score of 73.4% in binary behavior classification, outperforming the pipeline models (F1: 66.8% and 53.65%). On the German clinical dataset, the end-to-end model reached an even higher F1 score of 80.1%, again outperforming the pipeline (F1: 78.7%) and approaching the gold-annotated upper bound (F1: 92.9%). These results demonstrate that behavior classification benefits from direct extraction, and that our method generalizes across domains and languages. We release our code and dataset at : https://github.com/MaggieK410/Behavior_Extraction_from_Clinical_Interviews.git

1 Introduction

Accurate identification of behaviors and symptoms is essential for diagnosing developmental disorders such as Autism Spectrum Disorder (ASD), where behavior markers like repetitive movement, avoidant behaviors or absence of socially important behaviors are key diagnostic criteria (World Health Organization, 2023; American Psychiatric Association, 2013). However, existing tools for behavior and symptom identification rely heavily on qualitative analysis techniques (e.g., interviews, observations) that are time-consuming and subject to interpretation and whereby valuable information can be overlooked (Rutter et al., 2003; National Institute for Health and Care Excellence, 2023).

Although Event Extraction (EE) methods in NLP have been used in clinical contexts, such as extraction of symptoms and treatment decision in clinical records of medical disorders (Viani et al., 2020; Tung and Lu, 2016; Guzman-Nateras et al., 2022), general EE is ill-suited for ASD due to its heterogeneity and the nuanced presentation of behaviors and symptoms. Standard EE approaches identify general events, but not all events represent human behavior, particularly those that lack agentive or embodied action (Drury et al., 2022; Skinner, 1938). To support ASD diagnosis with behavior extraction, it is thus necessary to specify which events count as behaviors in the clinical sense. In this work, we frame binary behavior extraction as a specialized form of EE, and apply it to the analysis of actual clinical interviews in a novel corpus. Our contributions are threefold:
(1) We define a binary behavior classification scheme that embeds behavioral definitions within the TimeML framework.
(2) We present two new datasets for binary ASD behavior classification from events, annotated with our novel scheme: an English Reddit dataset from parents of autistic children and a German clinical interview corpus.
(3) We develop and compare pipeline and end-to-end models for behavior extraction both in English and German.
Our results show that end-to-end models trained directly on annotated behaviors outperform pipeline approaches.

In Section 2 we explore previous work on event extraction in clinical applications, in particular for behavior classification in ASD. We then describe the methods applied to both the English pilot study

Proceedings of the 16th International Conference on Computational Semantics, pages 153–165
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

and the German clinical data, including data preprocessing, the pipeline, and the direct classification approach for behavior analysis. In the subsequent Section 5 about the English pilot study, we detail datasets, considered models, training and results for EE, and behavior classification. We conclude Section 5 with learned lessons from the pilot study. Next, we focus on our experiments with German data in Section 6, which is structured in the same way as Section 5 and details the new dataset, models, training, results and discussion. We present our conclusions in Section 7 and the limitations of our work in Section 8.

2 Previous Work

2.1 Event Extraction and Clinical Applications

EE is a subtask of Information Extraction that focuses on identifying and categorizing events, defined as actions or occurrences situated in time and space. EE models typically extract event triggers and associated arguments (e.g., agents, objects, time). Benchmark datasets include ACE 2005 (LDC, 2005), which categorizes events across predefined types (e.g., Life, Movement, Conflict), and TimeBank (Pustejovsky et al., 2003a), which emphasizes temporal properties of events. TAC-KBP (Ellis et al., 2015) extends these with knowledge base population objectives. These corpora are primarily based on newswire or forum data, limiting their direct applicability to clinical language.

EE has successfully been applied in medical NLP, where it supports the extraction of symptoms, clinical events, and diagnostic information from unstructured texts such as electronic health records (EHRs) and clinical notes. For instance, EE has been used to detect negative emotions, thoughts, and symptoms from patient narratives and social media (Tung and Lu, 2016; Guzman-Nateras et al., 2022). However, such applications often focus on mood disorders, such as depression or anxiety, where symptom expressions are relatively homogeneous, for example, fatigue, reduced activity, and increased sleep. In contrast, symptoms of ASD can strongly vary by patient: While one patient can be highly verbal and socially eager, another patient can be non-verbal and seeking sensory input through self-stimulatory behaviour.

2.2 Event Extraction and ASD Detection

ASD presents unique challenges for automated analysis due to the heterogeneity of symptom presentation and atypical use of language.

Most digital tools for behavior and symptom identification in ASD remain underdeveloped. Standard NLP pipelines often fail to accommodate the idiosyncratic and context-dependent nature of autism-related behaviors (Calvo et al., 2017; Themistocleous et al., 2024).

Existing EE models assume relatively consistent linguistic patterns across populations, which makes them poorly suited for capturing the diverse behavioral descriptions in ASD, particularly from caregiver accounts (Zhang et al., 2022; Jurafsky and Martin, 2013). Due to data protection constraints, most text-based ASD studies rely on social media corpora (e.g., Reddit, Twitter) (Zirikly et al., 2019; Amir et al., 2019), which differ markedly from clinical interviews or third-party reports.

Although some work has focused on detecting discrete behaviors from text (Yates et al., 2017; Tadesse et al., 2019), these are typically surface-level behaviors and not grounded in a conceptual model of behavior relevant to developmental disorders (Skinner, 1938). Despite the similarity in the conception of events and behaviors, there exists a research gap in applying EE specifically to behavior detection for ASD within diagnostic settings.

2.3 Temporal Annotation with TimeML

TimeML is an annotation framework that supports fine-grained event labeling, including temporal categories (Pustejovsky et al., 2003b). Its application in mental health NLP has enabled the construction of patient timelines (e.g., tracking the duration of untreated psychosis from EHRs) (Viani et al., 2020). These tools help model behavioral onset and change over time, which is highly relevant for developmental disorders. While our work does not yet focus on temporal reasoning, TimeML's structured event taxonomy forms the basis of our behavior classification system.

2.4 Behavior Extraction and ASD

Our approach addresses this gap by adapting EE to behavior extraction, using third-party reports (e.g. transcripts of parent interviews) and applying a behavior-specific classification scheme grounded in TimeML, supporting the extraction of diagnostically relevant information for ASD.

3 Defining Behaviors in Terms of Events

Behaviors and events share key properties: they are observable, unfold over time, involve agents, and can be causally linked to outcomes. However, not all events describe behaviors. For behavior extraction in text, we require a more specific definition grounded in linguistic and psychological theory.

The TimeML annotation framework (Pustejovsky et al., 2003b) categorizes events into types such as *Occurrence* (actions that happen), *Perception* (sensory experiences), *Reporting* (communication acts), *Aspectual* (beginning, ending, or continuing another event), *I_Action/I_State* (intentions or mental states), and *State* (persistent conditions). These categories offer a rich base for distinguishing between behaviors and other event types.

Our behavior annotation follows a two-step process: (1) identify TimeML-style events in text; and (2) classify which of these constitute behaviors and which do not in a binary fashion. For instance, only agentive and embodied actions (e.g., a child "makes eye contact" or "repeats phrases") qualify as behaviors. In contrast, mental states or results of actions (e.g., "was upset", "was ignored") are excluded. Some event types like *State* and *Aspectual* never meet the definitional criteria for behavior as action by definition.

This filtered annotation is used to train both the pipeline and end-to-end behavior classification models. By grounding our behavior definition in the TimeML schema, we bridge the gap between generic event detection and clinically meaningful behavior identification.

4 Methods

We experiment with two datasets: a publicly available English Reddit dataset newly annotated for events and behaviors, and a novel German clinical interview dataset. The first enables comparison with existing methods, while the second provides real-world clinical insights. We discuss here the common aspects of the methods used in the two experiments.

4.1 Pre-processing

The pre-processing for both English and German data is identical and differs only by task and model. The BERT model classifies each token of the input, while generative models such as T5 and Phi3 get textual input with prompts and generate an output sequence. Since the generative models might produce outputs of different length than the input, we chose to set the maximum output length to be equal to length of the input sentence plus one additional token. This constraint promotes precision by excluding tokens beyond the input length from the loss calculation, thereby increasing the influence of earlier errors in the sequence.

For the EE task, both BERT and T5 receive a raw sentence as input. We embed the input sentence into one of two prompt templates with varying amounts of contextual information about events, and examples for Phi3 model (see Appendix A.1). The outputs are post-processed to ensure consistency for evaluation. The BERT token classification model outputs a predicted class label for each input token. The generative T5 and Phi-3 models produce event-tagged sentences that include event delimiters. An example of the raw input sentence, the desired event-tagged sentence and the token-wise classifications by BERT is given below:

Raw input sentence: Aber wir beginnen mal. (Translation: "But let's get started.")
Event-tagged sentence: Aber wir [ASPECTUAL] beginnen [END ASPECTUAL] mal.
Tokenized raw sentence:["Aber", "wir", "beginnen", "mal"]
List of token event classifications: [0, 0, 3, 0, 0]

Behavior classification can be approached either end-to-end, using the raw input sentence, or in a pipeline setting, using an event-tagged sentence as input. In behavior classification using BERT as a token classifier (BERT BC), each token is labeled as either not an event (number 2), an event but not a behavior (number 0), or an event and also a behavior (number 1). For the Phi3 behavior classification model (Phi3 BC), we add definition of behavior, extract the mentions in the tagged sentence and instruct it to classify each mention into behavior and non-behavior (see Appendix A.2). Phi3 outputs a list of mentions and their corresponding classifications. For evaluation, we disregard specific event categories by replacing them with "[EVENT]", "[END EVENT]" and "[EVENT, Bx]", since the end-to-end model can only classify into behavior and non behavior tokens. "[EVENT, Bx]" refers to the beginning of an event that is also a behavior, while "[EVENT]" marks the beginning of an event that is not a behavior. "[END EVENT]" is the end delimiter for both types of behaviors. This is illustrated in the following example:

Behavior annotated event-tagged sentence: Er [OCCURRENCE, Bx] spricht [END OCCURRENCE] mit dem Hund meiner Schwester Englisch. (Translation: "He speaks English with my sister's dog")
Event-tagged input sentence: Er [OCCURRENCE] spricht [END OCCURRENCE] mit dem Hund meiner Schwester Englisch.
Tokenized event-tagged sentence: ['Er', '[OCCURRENCE]', 'spricht', '[END OCCURRENCE]', 'mit', 'dem', 'Hund', 'meiner', 'Schwester', 'Englisch
Gold token map: [2, 2, 1, 2, 2, 2, 2, 2, 2, 2]

4.2 Behavior Classification: Pipeline vs. End-to-End

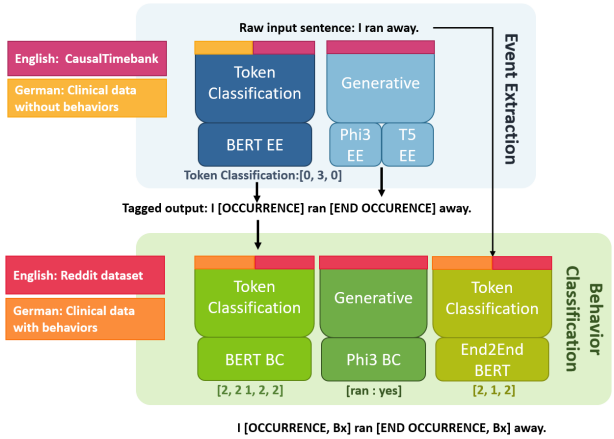


Figure 1: Overview of our experiment design with both English and German data.

For pipeline models, we consider only events as potential behaviors: we first extract events using the BERT EE token classification model, and then classify some of these events as behaviors in a second step with either BERT BC or Phi3 BC as the behavior classification model. By incorporating semantic information about event classes and mentions, the pipeline approach allows the behavior classifier to benefit from correlations between behaviors and specific event types. We investigate whether these benefits transfer equally across structurally different languages, such as German and English.

In the end-to-end model, each token in the raw input sentence is classified without any information about events. While the raw input provides no additional event information to the behavior classifier, avoiding a pipeline architecture reduces the risk of error propagation.

In a pilot study on publicly available english data, we compare generative models with token classifiers at both event extraction and behavior classification level in our pipeline. We contrast the pipeline approach with direct behavior classification on token level. The pilot study allows us to draw preliminary conclusions about which model types are successful before applying them to German clinical data. Figure 1 shows a visual overview of the pipeline and the direct approach to the behavior classification task and highlights, which models were trained with English data and which with German clinical data.

4.3 Post-Processing and Evaluation

Our evaluation for both tasks focuses on the mentions that are extracted from a model output. Evaluating EE models is inherently challenging due to the possibility of partial correctness, for example, extracting the correct text span but assigning the wrong event class (Peng et al., 2023; Zheng et al., 2021). Generative models such as Phi3 and T5 introduce additional complexity. While they are effective for tasks without strict output constraints, they are prone to hallucinating mentions or entire event classes.

In our EE evaluation, we employ F1, precision, and recall to evaluate the models' performances in identifying event classes, mentions, and spans. We extract mentions by aligning the model's token map outputs with the original sentences for BERT and extract mentions using regex from the tagged sentences generated by Phi3 and T5.

As mentioned in Section 4.1, we disregard specific event classes in the behavior classification task. Instead, we introduce the "Bx" addition ([EVENT, Bx]) to the class-neutral event delimiter [EVENT] to indicate that an event has been classified as a behavior. We then compare the extracted mentions, their spans, and the associated behavior classifications. Additionally, we compare the EE part of the pipeline to the end-to-end model by omitting the "Bx" addition and place a delimiter wherever the token map has a value that is not 2.

5 Pilots with Existing (English) Data

While developing the new German clinical dataset, we piloted our approach on an existing English dataset of non-clinical texts, which was newly annotated for behaviors and events by our clinical collaborators. This allowed us to evaluate different

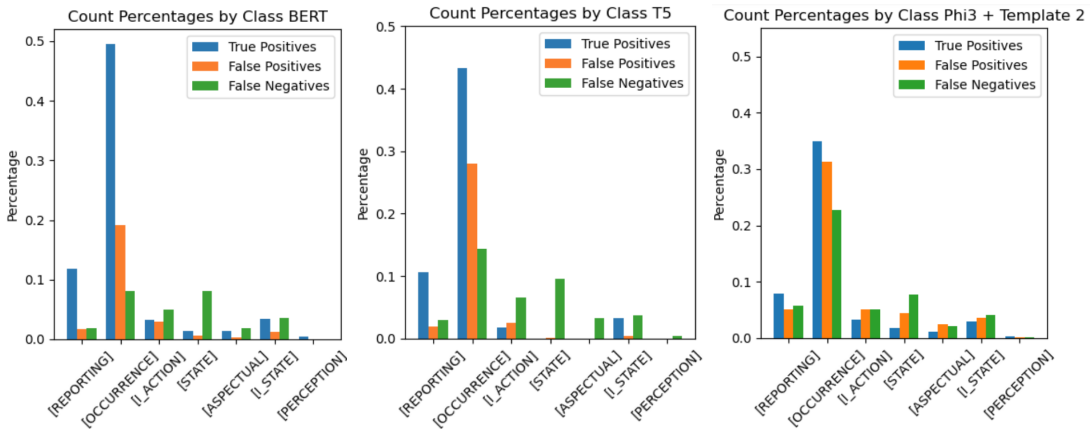


Figure 2: Normalized counts of false positives, false negatives, and true positives by class. We exclude "None" classifications, which refer to tokens that do not correspond to any event, since all models handle this majority class similarly. Phi3 shows the highest rate of false positives due to hallucinations and also the most false negatives, particularly in the *Occurrence* class.

methods in comparison to existing event extraction approaches. We report these preliminary experiments in this section.

5.1 TimeML Event Extraction

As a pilot experiment on English data, we evaluate a diverse set of model families to assess their suitability for downstream application to the German dataset. Specifically, we compare generative models Phi3-mini-128k-instruct (referred to as Phi3 hereafter)[1] and T5-base[2], with token classification model BERT-base-cased[3].

5.1.1 Data

For training and testing EE on English data we used CausalTimebank,[4] a freely available subset of the TimeML-annotated TimeBank dataset. We split the 6811 articles into 2655 sentences and produce train, test and validation sets with a 8:1:1 ratio (2123, 266, 266 sentences). All TimeML event classes are present in the dataset. These are *Occurrence*, *Reporting*, *I_Action*, *State*, *I_State*, *Aspectual* and *Perception*. This mirrors our German clinical dataset, which also includes all event classes.

5.1.2 Models and Training

We train the BERT-base-cased model for ten epochs on our pre-processed CausalTimebank dataset using the token classification objective with a learning rate of 2e-5. The generative models T5 and Phi3 are trained over 5 epochs. For Phi3 we set the learning rate to 2e-5 and the maximum length of the output to 1500 characters to accommodate the prompts. We set the learning rate to 5e-5 for T5. All models were trained on a single A40 GPU with 48GB RAM.

5.1.3 Results and Discussion

In Table 1 we report the performance of the models based on exact matches of mention, span and event class. We compared the weighted true positive, false positive, and false negative counts by event class for the BERT and T5 models with the Phi3 model using template 2 in Figure 2.

Model	Prec.	Rec.	F1
BERT-base-cased (BERT EE)	69.90%	72.19%	71.41%
T5	65.15%	56.12%	60.27%
Phi3 + template 1	72.61%	59.55%	65.08%
Phi3 + template 2	73.56%	65.63%	69.37%

Table 1: Recall, precision and F1 values for exact match for span, mention and event class for the models employed.

The results show BERT outperforming T5 and Phi3 in F1 and recall value. While Phi3 achieves slightly higher precision values then BERT and T5, but lower recall values lead to overall lower F1 values. The T5 model is outperformed by both Phi3 and BERT. The two template variants of Phi3 alter the F1 score by 4% and have a greater effect on recall, and consequently on the false negative rate, than on precision. This difference in performance highlights the importance of prompt engi-

[1] https://huggingface.co/microsoft/Phi-3-mini-128k-instruct

[2] https://huggingface.co/google-t5/t5-base

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

[4] https://github.com/paramitamirza/Causal-TimeBank

neering. Recent research (Shiri et al., 2024) shows that providing more event information improves model performance. However, this approach increases training time and memory costs. Reward functions have also been shown to enhance LLM performance in EE (Gao et al., 2024) and could be a starting point for future work.

Elevated levels of false positives in Figure 2 indicate that generation errors influence the results of T5 and Phi3. The generative models also exhibit higher false negative counts especially in the high occurring classes compared to BERT. BERT achieves the highest true positive rate for the most common class *Occurrence*. T5 underperforms on the *Occurrence* class and shows higher false positive rates across all classes, a pattern even more pronounced in the Phi3 results. Phi3 includes more false negatives in the *Occurrence* class, but has similar false negatives levels in the rarer classes, showing that the performance difference mainly stems from the most prevalent *Occurrence* class. Limiting output to the input length plus one token simplifies alignment and prevents drifting, but may lower the number of true positives in T5. When false positives are generated at the beginning of the sentence, the cutoff potentially eliminates correctly tagged mentions later in the sentence.

5.2 Behavior Classification

We train BERT BC and Phi3 BC as behavior classifiers on our English Reddit dataset and compare a pipeline approach with an end-to-end approach.

5.2.1 Data

Although our primary evaluation uses German clinical data, access to high-quality medical datasets is often limited. To train models for behavior annotation in English, we use the publicly available Reddit dataset[5] with posts collected between December 2022 and March 2024 from Autism related subreddits. These posts, primarily written by parents detailing their autistic children's behaviors and experiences, are shorter and less structured than professional consultations, but serve as a valuable resource for testing the abilities of models to extract useful information from third-party descriptions of behavior. Leveraging publicly available data, shows that our approach generalizes to non-clinical settings and may enable future cross-lingual analyses.

[5] https:\/\/huggingface.co\/datasets\/Osondu\/reddit_autism_dataset

Two clinical psychology experts, trained in the TimeML scheme, classified events and behaviors in 1,000 posts from raw text, creating a new English dataset used for the experiments in this section. We obtained a Fleiss kappa value of 0.53 for inter annotator agreement, which in the psychological literature is considered fair to good.

We consider the 743 sentences that contain events, and obtain a total of 2159 events from the annotated Reddit data. The data was split into train, test and validations splits with a ratio of 8:1:1 resulting in 216, 221 and 1722 mentions for test, validation and training, respectively.

This Reddit dataset, the first behavior classification dataset grounded in the psychological definition of behavior, will be released alongside this paper.

5.2.2 Models and Training

For the pipeline behavior extractor, we trained BERT for token classification (BERT BC) and Phi3 (Phi3 BC) for mention-level classification (see Appendix A.2) on our expert-annotated event sentences, using 10 epochs and a learning rate of 2e-5. For evaluation, we combine BERT BC with human-annotated events to estimate an upper bound, and use both Phi3 BC and BERT BC with events extracted by the BERT EE model from our previous experiment. We compare these pipeline models with an end-to-end BERT token classification model, predicting behaviors directly from the raw input.

5.2.3 Results and Discussion

We report precision, recall and F1 values for exact match of span and mention with and without behavior classification in Table 2. We also display an error analysis on token level using confusion matrices for each classification in Figure 3 for the two best performing pipeline models. Overall, the end-to-end BERT model outperforms the pipeline approach with a BERT EE model and a subsequent BERT BC or Phi3 BC behavior classifiers. The upper bound results using gold event annotations show that with a perfect event extraction model, a pipeline approach would significantly improve behavior classification over an end-to-end model. This suggests that the semantic information carried in the tagged events could enhance performance if captured accurately with the EE model. However, the performance gap between pipelines using gold versus predicted events illustrates the difficulty of

accurate event extraction and how errors in this step reduce the pipeline's overall effectiveness. Additionally, Phi3 BC performed poorly as a behavior classifier and introduced further errors, possibly due to the limited size of the Reddit dataset or a suboptimal prompt.

The confusion matrices indicate that models can learn to distinguish behaviors from non-behaviors, which indicates the presence of identifiable patterns that make behavior extraction statistically feasible.

With behavior classification

Model	Prec.	Rec.	F1
Gold EE + BERT BC	82.50%	82.50%	82.50%
End-to-end: BERT BC	**73.27%**	**73.61%**	**73.44%**
BERT EE+BERT BC	67.17%	66.50%	66.83%
BERT EE+Phi3 BC	54.56%	52.78%	53.65%

Without behavior classification

Model	Prec.	Rec.	F1
End-to-end: BERT BC	**85.25%**	**85.65%**	**85.45%**
BERT EE+BERT BC	83.73%	81.02%	82.35%

Table 2: Recall, precision and F1 value for exact match of span and mention with and without behavior classification on the English Reddit dataset.

5.3 Lessons Learned

Our experiments show that generative models are less suited for EE, as they often produce false positives and hallucinations that compromise performance and complicate evaluation. We compared a pipeline using BERT EE for event extraction and BERT BC or Phi3 BC for behavior classification with an end-to-end BERT model that labels tokens directly. We find that while the pipeline approach can outperform direct token classification under perfect EE, errors from the EE step accumulate and degrade performance. Additionally, since large clinical datasets are often unrealistic in real-world settings and BERT performs significantly better, we use BERT for downstream analysis on the German data. We conclude that token level behavior classification from raw input sentences performs best on the English dataset. To assess how well our approach generalizes across languages, we apply it to structurally different German data, by comparing a behavior classification pipeline (using both gold and BERT-extracted events) to an end-to-end BERT token classification model.

6 Experiments on German Clinical Data

Our main experiment involves the same steps as the pilot with English data, but this time applied to

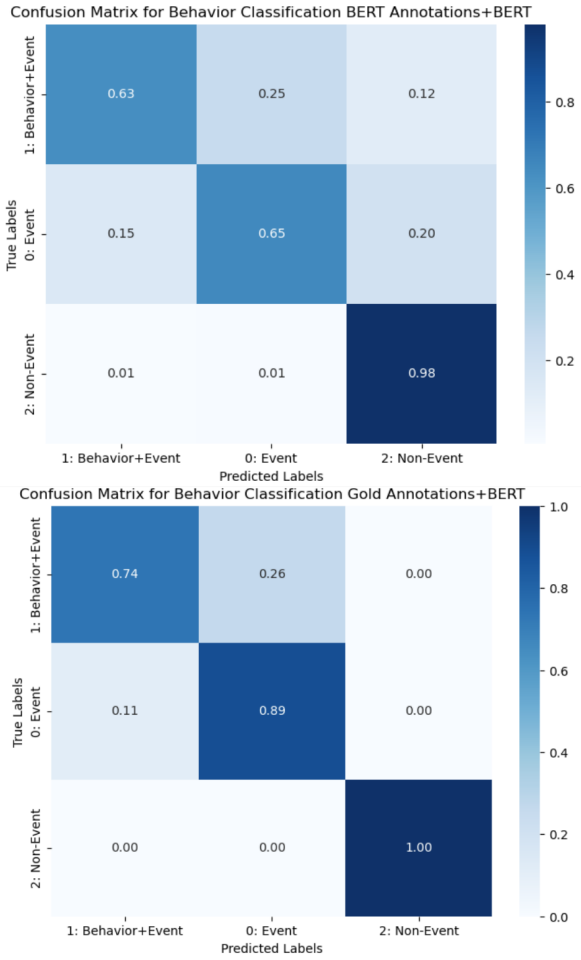


Figure 3: Normalized confusion matrices for behavior classification with BERT event annotations (top) and the gold annotations (bottom) on the Reddit dataset.

real clinical data in German.

6.1 A New Dataset

The English pilot study only includes newspaper articles from CausalTimebank (for EE) and non-clinical Reddit posts (for behavior extraction). For proper evaluation in a clinical setting, we create and annotate four transcribed sessions with parents of autistic children and qualified psychologists asking directed questions about the child's behavior and development using the ADI-R interview. This data was first TimeML annotated by the same clinical experts that annotated the English dataset, and subsequently behavior annotated using the same scheme used for the Reddit dataset. In total, the dataset contains 6566 events. We split our data by patient since we want to be able to generalize to other patients and simulate a realistic training environment. We select two patients for the training dataset, leading to 4,254 events, and the two remaining ones for test and validations set, with 1123

and 1189 events, respectively. We have enough events to train an EE model and a subsequent behavior classification model, as well as a end-to-end model on this data.

This second clinical dataset will also be made publicly available after publication.

6.2 Models and Training

Based on our experiments on English data, we select BERT-base-multilingual-cased[6] for the pipeline, and compare it with end-to-end classification from raw input. We do not use generative models, as we saw on the English data that they achieve lower performance compared to the token classification models. We prepare three distinct versions of the dataset for our experiments: (1) one with raw inputs and event-tagged outputs for training EE models; (2) one with raw inputs and behavior-tagged outputs for training the direct behavior classification model; and (3) one with event-tagged inputs and behavior-tagged outputs for training the behavior classification model using extracted event information. All models were trained for 5 epochs.

6.3 Results and Discussion

Table 3 reports the results for four setups: (1) the end-to-end BERT model, (2) a pipeline using BERT for event extraction (EE) and either BERT or (3) Phi3 for behavior classification, and (4) behavior classification on human-annotated sentences.

The best performing model from raw inputs is the end-to-end model, while the pipeline approaches suffer from error accumulation and performance decline. The upper bound for the subsequent behavior classification model is set by the EE component of the pipeline, which explains the weaker overall performance of the full pipeline.

The results on the clinical German data reflect the same pattern found in the English piloting experiment, which shows that the pilot on non-clinical, easily available data did yield valuable insights for this task that can be expanded to other languages.

However, we observed a notable improvement of $\sim$ 10% across all metrics in the German dataset compared to the English dataset. This is most likely due to the fact that our large clinical dataset contains three times as many events as the English Reddit dataset, enabling more reliable learning for the behavior classification model and resulting in better downstream performance, particularly evident in the BERT BC using human annotations. Additionally, its size of 6,566 events is of a similar scale to the 6,811 events in the CausalTimeBank dataset, allowing the German event extraction models to perform similarly to their English counterparts. A more detailed comparison between the German clinical dataset and CausalTimebank can be found in Figure 4. Since we split the German data by patient to ensure a more realistic clinical setting, event class distributions vary, potentially affecting the EE model's performance on the test set.

With behavior classification

Model	Prec.	Rec.	F1
Gold EE + BERT BC	92.93%	92.93%	92.93%
End-to-end: BERT BC	**79.13%**	**81.11%**	**80.10%**
BERT EE+BERT BC	77.48%	79.98%	78.71%

Without behavior classification

Model	Prec.	Rec.	F1
BERT EE	89.16%	92.03%	90.57%

Table 3: Recall, precision and F1 value for exact match of span and mention with and without behavior classification on the German clinical dataset.

7 Conclusions

We introduce a novel approach for identifying behaviors in text to support ASD diagnosis, by formulating behavior classification as a refinement of EE. Our analysis focuses on ASD behaviors which are described in third person by caretakers. Our approach was tested both on a newly created German dataset of clinical interviews with caretakers of potential ASD patients–to our knowledge, the first clinical German dataset with event and behavior annotations–as well on an existing, publicly available English dataset, which we also newly annotated using the same scheme.

Both of the new behavior classification datasets created for this work, and annotated by psychologists with extensive training in TimeML annotation, will be released.

Our results on both datasets show that the end-to-end model outperforms pipeline models that use an EE model followed by a behavior classifier, primarily due to error accumulation in the EE step. However, with optimal annotations in the EE step, a pipeline approach can outperform the end-to-end

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

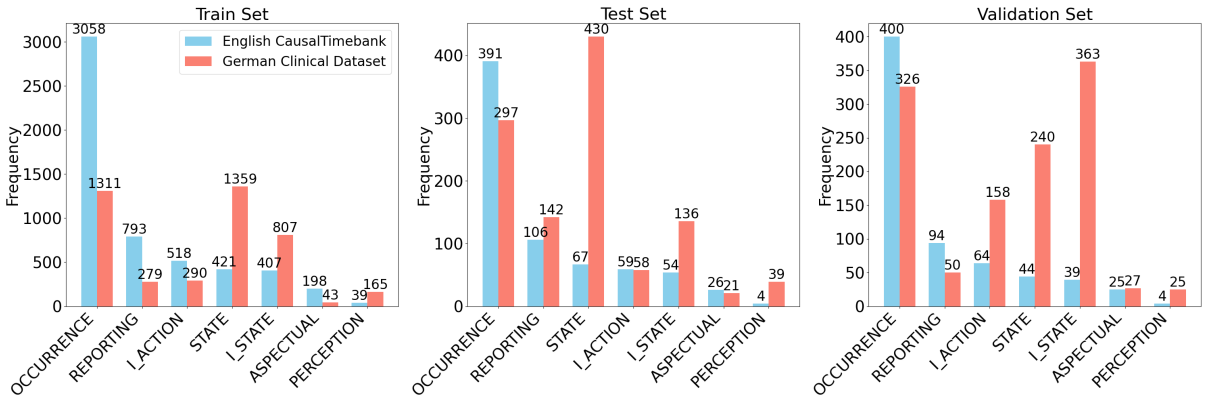


Figure 4: Frequencies of different event classes in the English CausalTimebank dataset and our German clinical dataset. The German data contains a more mixed profile, since we split by patients and not in a stratified way like we did in for CausalTimebank.

model. These results are similar in both English and German, suggesting that our approach is rooted in semantics of events and behaviors.

We can also infer from the results that the application of LLMs in the EE field is still challenging. Overall, Phi3 outperformed T5 with a slight margin, but different prompts for Phi3 had a notable impact on the performance indicating that prompt engineering needs to be further improved. On behavior classification, Phi3 performed notably worse than on EE, possibly because of the smaller dataset and an non-optimized prompt.

These results show that the extraction of behaviors conceptualized in terms of EE especially coupled with token classification has promise for the further development of this technology as well as implications for the development of clinical tool for disorders with idiosyncratic descriptions of behaviors. For example, we presented a prototype visual platform at HealTac2025, where clinicians can upload texts such as session transcripts, and our models extract and highlight events and behaviors in the submitted text.

8 Limitations

Our work explores the application of NLP in area of behavior classification in support of behavior analysis and is aimed at descriptions by parents of autistic children. Although we hope this work helps clinicians focus on important parts of the treatment and save time looking over transcripts and notes, we emphasize that these models do not have a perfect accuracy and are subject to not highlighting important parts of the text. Therefore, close analysis of the outputs by clinicians remains crucial.

Our work covers English and German data, but leaves many languages that might be syntactically different, and therefore more difficult to annotate, open for future work. Especially agglutinative languages might highlight the propagation of errors in EE. Additionally, we release the first German clinical dataset for behavior and event annotations, but there is currently a lack of large scale clinical datasets analyzing behavior and events in other languages.

9 Acknowledgements

Massimo Poesio's research was in part funded by ARCIDUCA, EPSRC EP/W001632/1. Ian Grant's research was funded through Queen Mary University London's Principal Scholarship.

References

American Psychiatric Association. 2013. *Diagnostic and Statistical Manual of Mental Disorders, 5th Edition (DSM-5)*. American Psychiatric Publishing, Arlington, VA. Autism Spectrum Disorder.

Silvio Amir, Mark Dredze, and John W. Ayers. 2019. Mental health surveillance over social media with digital cohorts. In *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*, pages 114–120, Minneapolis, Minnesota. Association for Computational Linguistics.

Rafael A. Calvo, David N. Milne, Mohammed S. Hussain, and Helen Christensen. 2017. Natural language processing in mental health applications using non-clinical texts. *Natural Language Engineering*, 23(5):649–685.

Benjamin Drury, Hugo Gonçalo Oliveira, and António de Araújo Lopes. 2022. A survey of the extraction and applications of causal relations. *Natural Language Engineering*, 28(3):361–400.

Joe Ellis, Jeremy Getman, Dana Fore, Neil Kuster, Zhiyi Song, Ann Bies, and Stephanie Strassel. 2015. Overview of linguistic resources for the tac kbp 2015 evaluations: Methodologies and results. *Theory and Applications of Categories*.

Jun Gao, Huan Zhao, Wei Wang, Changlong Yu, and Ruifeng Xu. 2024. Eventrl: Enhancing event extraction with outcome supervision for large language models.

Luis Guzman-Nateras, Viet Lai, Amir Pouran Ben Veyseh, Franck Dernoncourt, and Thien Nguyen. 2022. Event detection for suicide understanding. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1952–1961, Seattle, United States. Association for Computational Linguistics.

Daniel Jurafsky and James H. Martin. 2013. *Speech and Language Processing*, pearson new international edition edition. Pearson Education, London, UK.

LDC. 2005. ACE (Automatic Content Extraction) 2005 Multilingual Training Corpus. Linguistic Data Consortium, Philadelphia. LDC Catalog No.: LDC2006T06.

National Institute for Health and Care Excellence. 2023. Autism spectrum disorder in under 19s: support and management. Accessed: Jul. 04, 2023.

Hao Peng, Xiaozhi Wang, Feng Yao, Kaisheng Zeng, Lei Hou, Juanzi Li, Zhiyuan Liu, and Weixing Shen. 2023. The devil is in the details: On the pitfalls of event extraction evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9206–9227, Toronto, Canada. Association for Computational Linguistics.

James Pustejovsky, Jose Castano, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir Radev. 2003a. Timebank: Robust annotation of event and temporal expressions. In *Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5)*, pages 225–264, Tilburg, The Netherlands.

James Pustejovsky, José Castaño, Robert Ingria, Roser Saurí, Rob Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir Radev. 2003b. Timeml: Robust specification of event and temporal expressions in text. pages 28–34.

Michael Rutter, Ann LeCouteur, and Catherine Lord. 2003. *Autism Diagnostic Interview–Revised (ADI®-R)*. Western Psychological Services.

Fatemeh Shiri, Van Nguyen, Farhad Moghimifar, John Yoo, Gholamreza Haffari, and Yuan-Fang Li. 2024. Decompose, enrich, and extract! schema-aware event extraction using llms. *ArXiv*, abs/2406.01045.

B.F. Skinner. 1938. *The Behavior of Organisms: An Experimental Analysis*. Appleton-Century-Crofts, New York.

Mihretab Molla Tadesse, Hongmin Lin, Bo Xu, and Liang Yang. 2019. Detection of depression-related posts in reddit social media forum. *IEEE Access*, 7:44883–44893.

Charalambos K. Themistocleous, Maria Andreou, and Eleni Peristeri. 2024. Autism detection in children: Integrating machine learning and natural language processing in narrative analysis. *Behavioral Sciences*, 14(6):459.

Chun-Hung Tung and Wen-Hsiung Lu. 2016. Analyzing depression tendency of web posts using an event-driven depression tendency warning model. *Artificial Intelligence in Medicine*, 66:53–62.

Nicholas Viani, Judy Kam, Liang Yin, et al. 2020. Temporal information extraction from mental health records to identify duration of untreated psychosis. *Journal of Biomedical Semantics*, 11(2):1–10.

World Health Organization. 2023. *ICD-11: International Classification of Diseases, 11th Revision*. World Health Organization. Autism spectrum disorder (6A02).

Andrew Yates, Arman Cohan, and Nazli Goharian. 2017. Depression and self-harm risk assessment in online forums. *arXiv preprint arXiv:1709.01848*.

Tong Zhang, Anne M. Schoene, Shi Ji, and Sophia Ananiadou. 2022. Natural language processing applied to mental illness detection: a narrative review. *NPJ Digital Medicine*, 5(1):1–13.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2021. Revisiting the evaluation of end-to-end event extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4609–4617, Online. Association for Computational Linguistics.

Ayah Zirikly, Philip Resnik, Özlem Uzuner, and Kristy Hollingshead. 2019. CLPsych 2019 shared task: Predicting the degree of suicide risk in Reddit posts. In *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*, pages 24–33, Minneapolis, Minnesota. Association for Computational Linguistics.

A Appendix

A.1 EE prompt

The first prompt template (referred to as **template 1** in the paper) includes no extra information about the events and is similar to the T5 input, apart from it containing a simple instruction:

″<|user|>Your task is to extract the events in a sentence. There are 7 event types to consider: OCCURRENCE, I_ACTION, I_STATE, ASPECTUAL, REPORTING, STATE, PERCEPTION. In the following sentence, please extract all the events based on the above classes. Remember, there can be multiple events in a sentence:″

We add the sentence, followed by <|end|><|assistant|> as instructed on the model's webpage to generate the outputs. This prompt yielded the best results.
The second prompt (**template 2**) includes example annotations for each class and performed slightly worse than template 1.

″<|user|>Your task is to extract the events in a sentence. There are 7 event types to consider: OCCURRENCE, I_ACTION, I_STATE, ASPECTUAL, REPORTING, STATE, PERCEPTION.
EXAMPLES:
REPORTING:<|user|>He said that the volcano was spewing gases.<|end|> <|assistant|>He [REPORTING]said[END REPORTING] that the volcano was spewing gases.<|end|>

OCCURRENCE:<|user|>Two moderate eruptions shortly before 3 p.m. Sunday appeared to signal a larger explosion<|end|>
<|assisstant|> Two moderate [OCCURRENCE]eruptions[END OCCURRENCE] shortly before 3 p.m. Sunday appeared to [OCCURRENCE]signal[END OCCURRENCE] a larger [OCCURRENCE]explosion[END OCCURRENCE]<|end|>

I_ACTION:<|user|>Israel has been scrambling to buy more masks abroad.<|end|>
<|assistant|>Israel has been [I_ACTION]scrambling[END I_ACTION] to buy more masks abroad.<|end|>

STATE: <|user|>No injuries were reported over the weekend.<|end|>
<|assistant|>No [STATE]injuries[END STATE] were reported over the weekend<|end|>

I_STATE:<|user|>The agencies fear they will be unable to crack those codes to eavesdrop on spies and crooks.<|end|>
<|assistant|>The agencies [I_STATE]fear[END I_STATE] they will be unable to crack those codes to eavesdrop on spies and crooks.<|end|>

ASPECTUAL:<|user|>The volcano began showing signs of activity in April for the first time in 600 years.<|end|>
<|assistant|>The volcano [ASPECTUAL]began[END ASPECTUAL] showing signs of activity in April for the first time in 600 years<|end|>

PERCEPTION:<|user|>Witnesses tell Birmingham police they saw a man running.<|end|>
<|assistant|>Witnesses tell Birmingham police they [PERCEPTION]saw[END PERCEPTION] a man running.<|end|>

In the following sentence, please extract all the events based on the above class descriptions.″

After this, we add the desired sentence followed by <|end|><|assistant|>. The model performed slightly worse with this prompt. A possible explanation could be the lost in the middle problem, where elements in the middle of a long prompt are forgotten.

A.2 Behavior Classification Prompt

We experiment with only one prompt for behavior classification. It includes a psychological definition and three example sentences:

″Behavior in psychology is defined as: "That portion of an organism's interaction with its environment that is characterized by detectable displacement in space through time of some part of the organism and that results in a measurable change in at least one aspect of the environment"

Examples:
<|user|> My son is 5 years old & is said to have level 1 autism In this sentence, does "said" describe behavior?<|end|>
<|assistant|> said: yes<|end|>

<|user|> Key words I should be looking for on their websites that are green flags or red flags? In this sentence, does "looking" describe a behavior?<|end|>
<|assistant|> looking: no<|end|>

<|user|> He also likes books and reads books to himself in his own " In this sentence, do "likes" and/or "reads" describe behavior?<|end|>
<|assistant|> likes: no: yes<|end|>″

To integrate the sentences from the dataset, we extract the mentions and create a sentence listing them as in the example. We exclude the three example sentences from the dataset.

The Proper Treatment of Verbal Idioms in German Discourse Representation Structure Parsing

Kilian Evang, Rafael Ehren, Laura Kallmeyer
Heinrich Heine University Düsseldorf
Universitätsstr. 1, 40225 Düsseldorf, Germany
{kilian.evang,rafael.ehren,laura.kallmeyer}@hhu.de

Abstract

Existing datasets for semantic parsing lack adequate representations of potentially idiomatic expressions (PIEs), i.e., expressions consisting of two or more lexemes that can occur with either a literal or an idiomatic reading. As a result, we cannot test semantic parsers for their ability to correctly distinguish between the two cases, and to assign appropriate meaning representations. We address this situation by combining two semantically annotated resources to obtain a corpus of German sentences containing literal and idiomatic occurrences of PIEs, paired with meaning representations whose concepts and roles reflect the respective literal or idiomatic meaning. Experiments with a state-of-the-art semantic parser show that given appropriate training data, it can learn to predict the idiomatic meanings and improve performance also for literal readings, even though predicting the correct concepts in context remains challenging. We provide additional insights through evaluation on synthetic data.

1 Introduction

Meaning representations such as Minimal Recursion Semantics (Copestake et al., 2005), Abstract Meaning Representations (Banarescu et al., 2013) or Discourse Representation Structures (Kamp and Reyle, 1993) form a link between natural language and the realm of symbolic computation, including ontologies and logical reasoning. They have uses in tasks such as information extraction, dialogue systems, and computer-assisted study of natural language semantics (Sadeddine et al., 2024). Meaning representations have traditionally been constructed from text using rule-based precision grammars or combinations of statistical syntactic parsers and rule-based interpretation systems (Copestake and Flickinger, 2000; Curran et al., 2007). More recently, larger quantities of annotated sentence-meaning pairs have made it possible to perform accurate data-driven text-to-meaning parsing (semantic parsing) and meaning-to-text generation (e.g., Flanigan et al., 2014; van Noord et al., 2020; Wang et al., 2023).

Decomposable verbal idiom: *Don't **spill** the **beans**!*

	x e
¬	**spill.v.05**(e) Agent(e, hearer)
	Theme(e, x) **secret.n.01**(x)

Non-decomposable verbal idiom:
*Are you **pulling** my **leg**?*

e
pull_the_leg_of(e) Agent(e, hearer)
Theme(e, speaker)

Literal occurrence of a verbal potentially idiomatic expression: *They like **playing games** on the PlayStation 2.*

x e f y z
person.n.01(x) like.v.02(e) Experiencer(e, x)
Stimulus(e, f) **play.v.01**(f) Agent(f, x)
Theme(f, y) Instrument(f, z) **game.n.01**(y)
entity.n.01(z) Name(z, "PlayStation 2")

Figure 1: Discourse representation structures for three sentences, containing different occurrences of potentially idiomatic expressions (PIEs). The bolded words are the components of the PIEs, the bolded concepts express their meanings in the respective context.

Datasets that have been constructed using computational grammars typically have a more or less strong built-in assumption that each occurrence of a content word is associated with exactly one occurrence of a *concept* (i.e., of a word sense from an ontology such as WordNet; Fellbaum, 1998). Furthermore, one typically assumes that while lex-

Proceedings of the 16th International Conference on Computational Semantics, pages 166–175
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

emes can be ambiguous, their senses do not depend on co-occurrence with specific other lexemes.

These assumptions break down in the case of *phrasemes* or *multiword expressions* (MWEs), i.e., combinations of two or more words expressing a single sense (e.g., *pull someone's leg*: `kid.v.01`), or being associated with different but specific senses when occurring together (e.g., *spill the beans*: `talk.v.04` and `secret.n.01`). MWEs occur in a variety of forms (Baldwin and Kim, 2010). In this paper, we focus on *verbal* MWEs, i.e., MWEs whose syntactic head is a verb. In particular, we focus on the subtype of *verbal idiom*. Ramisch et al. (2018) define verbal idioms (VIDs) as MWEs with at least two lexicalized components including the head verb and at least one of its dependents, excluding special cases like light verb constructions, verb-particle constructions, inherently adpositional verbs or inherently reflexive verbs. Following Nunberg et al. (1994), we further distinguish two subtypes of verbal idioms: *decomposable VIDs* such as *spill the beans* where the lexicalized components still have individual meanings even though they are specific to the combination, and *nondecomposable VIDs* such as *pull someone's leg* where all lexical components express a single concept together. Note also that even when two or more lexemes can form a VID together, they can still occur in the same syntactic configuration with a literal, non-idiomatic, compositionally derivable meaning. For example, the phrase *playing games* can occur with an idiomatic but also with a literal meaning. We are therefore dealing with *potentially idiomatic expressions* (PIEs; Haagsma, 2020) with both idiomatic and literal occurrences. It should be noted that PIE occurrences need not be contiguous but exhibit syntactic flexibility as in *the beans were spilled* or *the games that we played*.

In the context of semantic parsing, PIEs present specific challenges: 1) on encountering a PIE, the parser has to decide whether it indeed has the idiomatic meaning in this context, and 2) if so, it must produce the correct meaning representation, meaning one or more concepts that are specific to the idiom, and no additional concepts for additional components of non-decomposable idoms (see Figure 1).

In this paper, we demonstrate that existing semantic parsers for discourse representation structures underperform on sentences containing literal and idiomatic PIEs. We also show a way to remedy this situation. To this end, we combine two semantically annotated resources, the Parallel Meaning Bank (PMB; Abzianidze et al., 2017, 2020) and the dataset of Ehren et al. (2024), to obtain a corpus of German sentences containing literal and idiomatic occurrences of PIEs, annotated with meaning representations that reflect the correct meaning in context (Section 2). We then show that enriching the training data of a DRS parser with such data improves its performance on sentences containing idiomatic occurrences of PIEs, and in some cases its performance overall. Nevertheless, it remains challenging for the parser to reliably distinguish between literal and idiomatic uses, and also to choose the correct concepts for idioms (Section 3). We provide further insights with an evaluation on synthetic data (Section 4). We conclude in Section 5.

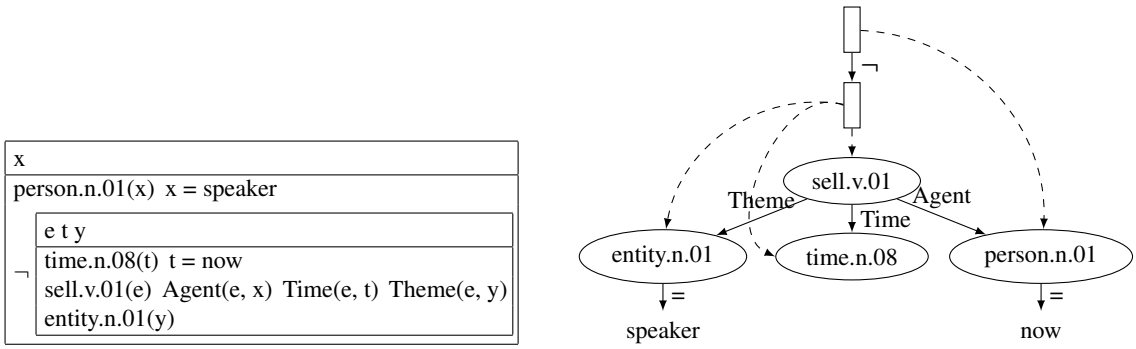
Besides these experimental designs and findings, our contributions include several reusable datasets which will be released upon publication, including an adjudicated version of Ehren et al.'s semantically annotated idiom dataset, an accordingly reannotated version of sentences containing PIEs in the Parallel Meaning Bank, and a synthetic dataset containing the annotated idioms isolated in canonical form, annotated with meaning representations.

2 Data

2.1 The Parallel Meaning Bank

The Parallel Meaning Bank (PMB; Abzianidze et al., 2017, 2020) is a partially parallel corpus of English, German, Italian, and Dutch texts, annotated with discourse representation structures (DRS) following Discourse Representation Theory (Kamp and Reyle, 1993), including word senses, semantic roles, discourse connectives, scope, coreference, etc. The annotations were created by an NLP pipeline and hand-corrected by human annotators. Completely checked documents have the status "gold", partially checked ones, "silver", and unchecked ones, "bronze". Even silver and bronze documents have been shown to be useful for training data-driven DRS parsers (van Noord et al., 2018).

In the PMB, a *document* consists of one or more sentences, paired with one DRS. Traditionally, DRSs are drawn as boxes as shown in Figure 2a. The top part of a box contains the *discourse referents*, which represent events, things, and other entities. The bottom part contains *conditions*, including a *concept condition* for each discourse ref-



(a) (b)

```
person.n.01 EQU speaker              % I am not
NEGATION <1
time.n.08 EQU now
sell.v.01 Agent -2 Time -1 Theme +1 % selling
entity.n.01                          % anything
```

(c)

Figure 2: DRS for the sentence "I am not selling anything" in (a) box notation, (b) graph notation, and (c) sequence notation. The sequence notation additionally shows concept-token alignment information. Adapted from Wang et al. (2023).

erent, saying what type of entity it is, *relation conditions* encoding semantic roles and other relations between entities, *equality conditions* linking referents to discourse constants such as `speaker` or `now`, and *complex conditions* consisting of a logical connective such as negation and one or two embedded boxes. DRSs can also be represented as *discourse representation graphs* (DRGs) as shown in Figure 2b. Here, boxes and discourse referents are represented as nodes. Referent nodes are labeled with their concepts. Box nodes have outgoing edges to the introduced referents and complex conditions, the latter labeled with discourse connectives. Relation conditions are encoded as edges between the referent nodes. Constants are encoded as nodes with incoming edges labeled =. Finally, this graph structure can be linearized as a sequence of tokens (Bos, 2023) as shown in Figure 2c. Here, nodes are encoded by their label, and edges are encoded following their source node by their label followed by a *pointer* indicating the relative position of the sink node. Concept nodes are aligned to the natural-language tokens that evoke them, also shown in Figure 2c.

2.2 German Verbal PIE Data

Ehren et al. (2024) argue that idioms are underrepresented in the gold part of the PMB, and they released a dataset of 6 187 sentences from the PMB's German part that contain verbal potentially idiomatic expressions (PIEs), annotated for whether the instance is idiomatic, and if so, for its sense, assigned roles, and, in the case of decomposable idioms, internal senses and roles. The following are examples of the annotations in this dataset. (1) is a PIE annotated as literal, (2) is a decomposable idiom annotated with senses and internal and external roles, and (3) is a non-decomposable idiom annotated with a sense and external roles.

(1) Tom lag bewusstlos auf dem Operationstisch
Tom lay unconscious on the operating table
PIE: auf dem Tisch liegen ("to be available; shown, offered")
Reading: literal

(2) Ich_[Theme] sitze_[be.v.01] im_[] selben_[]
I sit in the same
Boot_[Attribute]_[situation.n.02] wie du
boat as you
PIE: im selben Boot sitzen ("to be in the same boat")
Reading: idiomatic

(3) Tom_[Theme] kämpft, um über_[] die_[]
Tom fights, to over the
Runden_[] zu kommen_[survive.v.03]
rounds to come
PIE: über die Runden kommen ("to support oneself")
Reading: idiomatic

2.3 Adjudication

We extracted from Ehren et al.'s dataset the 2 204 sentences with a PIE annotated by at least one annotator as idiomatic, and thus with a semantic an-

Table 1: Four types of adjudication decisions, with examples. Struck out lines represent annotations that were discarded in favor of the other annotation. Struck out (underlined) spans represent parts of selected annotations that were removed (added) by the adjudicator. Annotations without struck out or underlined spans represent annotations that were approved unchanged in adjudication, shown here for comparison.

Consistent concepts and roles per sentence and per PIE type:
Das abgestürzte Flugzeug_[Patient] ging_[go_up.v.06] in_[] Flammen_[] auf_[]
~~Das abgestürzte Flugzeug_[Patient] ging_[go_up.v.01] in_[] Flammen_[] auf_[]~~
"The crashed plane went up in flames"
Die ,, Hindenburg_[Patient] " ging_[go_up.v.~~01~~06] plötzlich in_[] Flammen_[] auf_[]
"The 'Hindenburg' suddenly went up in flames"
Marking negation as part vs. not part of the idiom:
Käse_[Theme] und andere Milchprodukte_[Theme] bekommen_[agree.v.06] mir_[Attribute] nicht_~~[]~~
"Cheese and other dairy products do not agree with me"
Ich_[Experiencer] kann_[] sie_[Stimulus] nicht_[] ausstehen_[loathe.v.01]
"I cannot stand her/them"
Treatment of auxiliary (including modal) verbs as not head of clause:
Ich_[Experiencer] kann_[~~interest.v.01~~] mit_[] diesem Test_[Stimulus] nichts anfangen_[interest.v.01]
"I am not interested in this text" (lit. "I cannot begin anything with this text")
Treatment of adjective copula and auxiliary sein *as not part of idiom:*
Du_[Agent] musst auf alles_[Beneficiary] gefasst_[prepare_for.v.01] sein_~~[prepare_for.v.01]~~
"You have to be prepared for anything"

notation. This results in a total of 4 600 annotations (the number of annotations per sentence is slightly above 2), across 957 different PIEs. The first author went through the dataset manually and resolved divergent annotations according to Ehren et al.'s annotation manual. We also made sure that the same PIE was annotated consistently across occurrences, using the same WordNet sense and the same VerbNet roles for corresponding arguments. Because it was a frequent source of disagreement and affects automatic combination with the PMB data through word-concept alignment information (see next section), we made special adjudication passes to ensure conformance with the annotation guidelines wrt. the treatment of copulas, auxiliary verbs, and negation words like *nicht* "not". Examples are shown in Table 1.

2.4 Combining the Annotations with the PMB Data

The resulting unique annotations were automatically combined with the PMB 5.1.0, matching the annotations by sentence, using the alignment between tokens and concepts provided with the PMB. Examples are shown in Figure 3. For tokens annotated with a sense, the corresponding node was relabeled with that sense. For tokens annotated with a role, the incoming edge was relabeled with that role. For tokens annotated with an empty pair of brackets, the corresponding node and its incoming and outgoing edges were removed. As a result, we obtained 2 186 reannotated sentence-DRS pairs with idiomatic readings of PIEs.

2.5 Datasets

We prepared the following datasets for training and evaluating DRS parsers:

$\mathcal{I}$: the 2 186 automatically reannotated sentences containing idiomatic PIE instances, as described in Section 2.4.

$\mathcal{L}$: the 455 sentences marked by at least one annotator in Ehren et al.'s dataset as containing a literal reading of a PIE.

Note that both datasets may contain errors, as most of them are "bronze" or "silver", and reannotation only fixes the annotation of the PIE instance.

3 Targeted Training on PIE Instances

We assess the performance of a seq2seq parser on PIEs, comparing four different training conditions: training on the unmodified PMB data (baseline), adding available PIE instances into the training data (enhanced), adding a balanced mix of literal and idiomatic PIE instances into the training data (bal-

Er_[Experiencer] **schwimmt**_[buck.v.02] **gegen**_[] **den**_[] **Strom**_[Stimulus]_[trend.n.01]
"He bucks the trend" (lit. "He swims against the tide")

```
male.n.02                                                          % Er
~~schwimmt.v.01~~buck.v.02 ~~Agent~~Experiencer -1 Time +1 ~~Location~~Stimulus +2 % schwimmt gegen den
time.n.08              EQU now                                     %
~~tide.n.01~~                                                      % Strom.
```

Sie **steckt**_[despair.v.01]_[Experiencer] **den**_[] **Kopf**_[] **in**_[] **den**_[] **Sand**_[]
"She despairs" (lit. "She puts the head into the sand")

```
female.n.02                                                          % Sie
~~steckt.v.01~~despair.v.01 ~~Agent~~Experiencer -1 Time +1 ~~Theme +3 Location +4~~ % steckt
time.n.08                EQU now                                     %
~~female.n.01~~                                                     % den
~~head.n.01~~                ~~Participant -1~~                     % Kopf in den
~~sand.n.01~~                                                       % Sand.
```

Figure 3: Automatic combination of semantic idiom annotations with the PMB data via concept-token alignment. The meaning representations are discourse representation structures in sequence notation (cf. Section 2.1). Struck out (underlined) spans represent parts of the meaning representation that were removed (added) compared to the original PMB data. Note that some of the replaced senses, such as `steckt.v.01` or `schwimmt.v.01`, would be incorrect even in a literal reading, since they are not WordNet senses but artifacts of the bootstrapping process for the German DRS data.

anced), and weighing PIE instances more strongly than other training instances (balanced×4).

3.1 Model and Evaluation Metric

We use the seq2seq parser of Wang et al. (2023) as implemented by Zhang et al. (2024), with the pre-trained ByT5 language model (Xue et al., 2022). We further pre-train on PMB gold, silver, and bronze data for 3 epochs, then fine-tune on gold data (plus PIE data) for 10 epochs. We also follow these papers by using Smatch (Cai and Knight, 2013), adapted to DRS, as the evaluation metric.

3.2 Data Splits

We split $\mathcal{I}$ and $\mathcal{L}$ randomly into five equal parts and use a different part in each run for testing, reporting results as the median of five runs. We call this part $\mathcal{I}_{test}$ ($\mathcal{L}_{test}$) and the remainder $\mathcal{I}_{train}$ ($\mathcal{L}_{train}$).

For pre-training the **baseline** model, we use the PMB 5.1.0 German bronze, silver, and gold training portions, but with sentences in $\mathcal{I}_{test}$ and $\mathcal{L}_{test}$ removed. For fine-tuning the baseline model, we use the PMB 5.1.0 German gold training portion, which does not overlap with $\mathcal{I}_{test}$ or $\mathcal{L}_{test}$.

For pre-training the **enhanced** model, we use the same pre-training data as above except that sentences in $\mathcal{I}_{train}$ have their annotations replaced by the modified ones. For fine-tuning, we additionally add $\mathcal{I}_{train}$ and $\mathcal{L}_{train}$ to the fine-tuning data.

For fine-tuning the **balanced** model, we do the same but add $\mathcal{L}_{train}$ five times so that the count of idiomatic and literal training instances is approximately equal.

For fine-tuning the **balanced×4** model, we again multiply all the idiomatic and literal training instances by 4, thus weighting idiomatic PIE instances 4 times and literal ones 20 times as heavily as the standard training data. The value 4 was found in preliminary experimentation to improve accuracy for parsing idioms compared to 2 and to be on par with 8.

We then evaluate 1) on the PMB 5.1.0 standard gold test and dev sets; 2) on $\mathcal{I}_{test}$ and various subsets, *viz.* sentences with idioms seen in training, sentences with idioms not seen in training, and sentences sampled from shortest to longest to have the same mean length (in characters) as the standard test set; 3) on $\mathcal{L}_{test}$. The output DRSs are evaluated against the corresponding DRSs in the test sets using the Smatch metric.

3.3 Results

Results are shown in Table 2. We see that, compared to the baseline model, the enhanced model improves scores significantly even on the standard test and dev sets. This could be due to additional data helping even when it is not gold and does not directly address phenomena found in the test set. We see that compared to the standard dev and test sets, both models perform much worse on sen-

	baseline	enhanced	balanced	balanced×4
standard test	.815	.828*	.824*	.810
standard dev	.827	.835*	.832*	.819
idiomatic	.520	.572*	.567*	.606*
idiomatic seen	.530	.580*	.568*	.604*
idiomatic unseen	.518	.550*	.522*	.555*
idiomatic short	.614	.679*	.670*	.739*
literal	.650	.642	.658	.613

Table 2: Performance comparison of different models on the PMB 5.1.0 official test/dev data, on sentences with idioms, on sentences with seen and unseen idioms, on short sentences with idioms, and on sentences with literal PIE occurrences. Scores are macro-averages (mean) over the Smatch scores of the sentences; averaged (median) over five runs, each with a different test fold of idiom and literal sentences. * indicates statistically significant improvement over the baseline ($p \leq 0.05$) according to a permutation test (Dror et al., 2018).

tences containing PIEs and especially idiomatic occurrences, even when we downsample the latter to only contain idioms seen in training, or to contain only short sentences. This is partly due to idioms being challenging to handle, but also points to the test sentences' bronze/silver status, which we address in Section 4. But we also see that the enhanced model, additionally trained on reannotated idiomatic instances as well as literal PIE instances, performs several percentage points better on the idiomatic instances. As may be expected, the improvement on idioms that have not been seen in training is comparatively small. Performance on literal PIE instances is worse than the baseline model, suggesting that the enhanced model is biased towards idiomatic readings. This is not very surprising given the much larger size of $\mathcal{I}$ compared to $\mathcal{L}$. By contrast, the balanced model avoids degradation on literal instances and in fact improves accuracy (though not significantly) while also still improving over the baseline significantly on all other test sets, albeit slightly less than the enhanced model. It is worth noting that targeted training on PIEs can thus maintain performance on literal instances although literal interpretations are the default case in the standard training data. Finally, the balanced×4 model achieves the best accuracy on idiomatic readings, but performs worse than the baseline on literal readings, and also degrades on the standard test sets.

4 Evaluation on Synthetic Data

The data in $\mathcal{I}$ and $\mathcal{L}$ is based on bronze and silver documents in the PMB and thus contains errors, even though concepts and roles representing the meanings of idioms have been automatically fixed in $\mathcal{I}$. The above experiments thus give a somewhat misleading view of the parsers' performance. To better understand the performance on idioms without any unrelated error sources, we perform an evaluation on a synthetic 'test set' of minimal sentences containing idioms, paired with DRSs that are correct by construction, assuming the idiomatic reading.

4.1 Construction of the Synthetic Dataset

We went through the adjudicated idiom sentences and reduced each idiom to a natural-language canonical form, similar to Odijk and Kroon (2024). In our case, canonical forms are main clauses, manually chunked, and decorated with senses and roles. DRSs can be automatically generated from the canonical forms by mapping placeholder words such as *etwas* "something", *irgendwie* "somehow", *irgendwohin* "somewhere", or *jemand* "somebody" to arbitrary fillers; we simply use general concepts representing the meaning of the placeholders, *viz.* `entity.n.01`, `manner.n.01`, `location.n.01`, and `person.n.01`, respectively. For example:

(4) [Jemand]$_{\text{Patient}}$ kommt$_{\text{die.v.01}}$ [ums Leben]
somebody comes around the life
"Somebody dies"
`person.n.01 die.v.01 Patient -1 Time +1 time.n.08 EQU now`

(5) [Etwas]$_{\text{Stimulus}}$ geht$_{\text{annoy.v.01}}$
something goes
[jemandem]$_{\text{Experiencer}}$ [auf die Nerven]
somebody on the nerves
"Something annoys somebody"
`entity.n.01 annoy.v.01 Stimulus -1 Time +1 Experiencer +2 time.n.08 EQU now person.n.01`

	baseline	enhanced
dev	.695	.767*
test	.689	.765*
dev decomposable	.706	.722*
dev non-decomposable	.678	.752*
test decomposable	.720	.736*
test non-decomposable	.692	.757*

Table 3: Results on synthetic test data. * indicates statistically significant improvement over the baseline ($p \leq 0.05$) according to a permutation test (Dror et al., 2018).

(6) [Etwas]$_{\text{Patient}}$ geht$_{\text{come.v.04}}$ [in Erfüllung]$_{\text{true.a.01}}$
something goes into fulfillment
"Something comes true"
`entity.n.01 come.v.04 Patient -1 Time +1 Result +2 time.n.08 EQU now true.a.01`

We obtained 890 sentence-DRS pairs in this way and split them randomly into an even-sized development set and test set.

4.2 Experiments

We use the baseline model and the enhanced model from Section 3.2. Now, instead of 5-fold cross validation, we add all of $\mathcal{I}$ and $\mathcal{L}$ to the fine-tuning data and evaluate on the synthetic data.

4.3 Results

Results are shown in Table 3. With the syntactic structure and the interpretation of arguments trivial in the synthetic data, scores now almost exclusively reflect the model's ability to map the idiom to the correct sense(s) and roles. Again, the enhanced model does significantly better than the baseline model.

We also see that the baseline model does better on decomposable than on non-decomposable idioms. This makes sense, as the correct interpretations of decomposable idioms are structurally closer to literal readings, with two senses rather than one. In the enhanced model, this is reversed: it does better on non-decomposable idioms. This shows that the model has learned to predict the non-canonical structure of non-decomposable idioms. The better scores are probably also due to one sense being statistically easier to predict correctly than two, and to the stronger representation of non-decomposable idioms in our training data.

We show here some examples that the baseline model parses wrongly and the enhanced model parses correctly:

(7) Jemand macht sich über jemanden
Somebody makes themself about somebody
lustig
funny
"Somebody mocks somebody"
Baseline: `person.n.01 make.v.01 Agent -1 Time +1 Product +2 Theme +3 time.n.08 EQU now male.n.02 person.n.01 funny.a.01 AttributeOf -1`
Enhanced: `person.n.01 mock.v.01 Agent -1 Time +1 Theme +2 time.n.08 EQU now person.n.01`

(8) Jemand setzt jemanden über etwas in
Somebody sets somebody about something in
Kenntnis
knowledge
"Somebody informs somebody"
Baseline: `person.n.01 put.v.01 Agent -1 Time +1 Theme +2 Theme +3 time.n.08 EQU now person.n.01 entity.n.01`
Enhanced: `person.n.01 inform.v.01 Agent -1 Time +1 Recipient +2 Topic +3 time.n.08 EQU now person.n.01 entity.n.01`

(9) Etwas geht vor sich
Something goes before itself
"Something happens"
Baseline: `entity.n.01 go.v.01 Theme -1 Time +1 time.n.08 EQU now`
Enhanced: `entity.n.01 happen.v.01 Theme -1 Time +1 time.n.08 EQU now`

(10) Jemand weiß etwas zu schätzen
Somebody knows something to value
"Somebody appreciates something"
Baseline: `person.n.01 know.v.01 Experiencer -1 Time +1 Stimulus +2 time.n.08 EQU now entity.n.01 appreciate.v.01 Agent -4 Theme -1`
Enhanced: `person.n.01 appreciate.v.01 Experiencer -1 Time +1 Stimulus +2 time.n.08 EQU now entity.n.01`

As for the effect of the frequency of an idiom in the training data, a scatterplot (Figure 4) shows that although high scores on the synthetic data are already achieved with as little as one training example, reliably decent scores are only seen around 10 or more training examples.

5 Conclusions, Limitations, and Future Work

Potentially idiomatic expressions (PIEs) present a special challenge in semantic parsing due to their

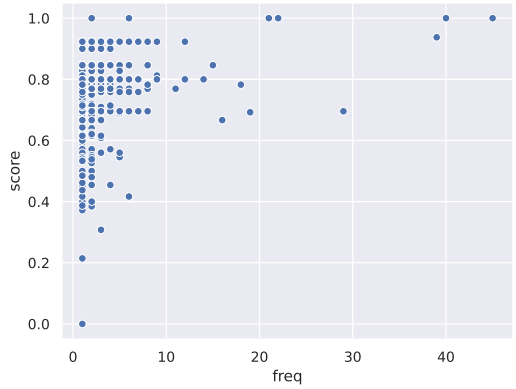


Figure 4: Scatterplot of idiom frequency in the training data against smatch score in the synthetic development data.

idiomatic meanings in some contexts, often with a single concept expressed by two or more content words. For an existing state-of-the-art system for parsing German text to discourse representation structures, we have shown that it struggles with sentences containing verbal PIEs more than with the average sentence. We have also shown that the gap can be partially closed without changing the parser's sequence-to-sequence architecture, simply by injecting sentences with PIEs into the training data, where sentences with idiomatic readings have been reannotated to reflect these.

Our contributions also include an adjudicated version of Ehren et al.'s German semantically annotated verbal PIE dataset, a correspondingly reannotated version of 2 186 German sentences in the Parallel Meaning Bank which we intend to submit for inclusion into the next release of the PMB, and a synthetic dataset of 890 idioms in isolated canonical form, with corresponding meaning representations.

There are two main limitations: the first concerns evaluation. Because we had only partially corrected test data at our disposal, we still only have an approximate picture of how accurately models handle PIEs. We partially addressed this by evaluating on synthetic data, but future work should aim to get an accurate picture on idiom semantic parsing accuracy on real data.

The second limitation applies to semantic parsing in general: meaning representations are expensive to annotate, thus the training data is limited in quantity and quality, with model training having to rely on partially corrected data. Although we achieved improved scores on sentences containing idioms, in many cases the models still struggle to pick the correct sense. As performance grows on idiomatic instances, it goes down on literal ones, suggesting that models seem to prefer one or the other and struggle with distinguishing between literal and idiomatic occurrences in context.

Future work should build on our synthetic dataset by using it not just for testing but also for training, automatically generating from the canonical forms sentences more varied in clause type, embedding complexity, fillers for placeholders, negation, modality, tense, etc. In addition, it may be worth making the decision between idiomatic and literal readings explicit and delegating it to a specialized model.

Acknowledgments

We would like to thank the anonymous reviewers for their feedback. We would also like to thank our annotators for their work. This work was carried out in the MWE-SemPrE project funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), project number 467699802. Computational infrastructure and support were provided by the Centre for Information and Media Technology at Heinrich Heine University Düsseldorf.

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Lasha Abzianidze, Rik van Noord, Chunliu Wang, and Johan Bos. 2020. The parallel meaning bank: A framework for semantically annotating multiple languages. *Applied mathematics and informatics*, 25(2):45–60.

Timothy Baldwin and Su Nam Kim. 2010. Multiword expressions. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing*, second edition edition, pages 267–292. CRC Press, Boca Raton.

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.

Johan Bos. 2023. The sequence notation: Catching complex meanings in simple graphs. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 195–208, Nancy, France. Association for Computational Linguistics.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Ann Copestake and Dan Flickinger. 2000. An open source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

Ann A. Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3:281–332.

James Curran, Stephen Clark, and Johan Bos. 2007. Linguistically motivated large-scale NLP with C&C and boxer. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 33–36, Prague, Czech Republic. Association for Computational Linguistics.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Rafael Ehren, Kilian Evang, and Laura Kallmeyer. 2024. To leave no stone unturned: Annotating verbal idioms in the Parallel Meaning Bank. In *Proceedings of the Joint Workshop on Multiword Expressions and Universal Dependencies (MWE-UD) @ LREC-COLING 2024*, pages 115–124, Torino, Italia. ELRA and ICCL.

Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT Press.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the Abstract Meaning Representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Baltimore, Maryland. Association for Computational Linguistics.

Hessel Haagsma. 2020. *A Bigger Frish to Fry: Scaling up the Automatic Understanding of Idiomatic Expressions*. Ph.D. thesis, Rijksuniversiteit Groningen.

Hans Kamp and Uwe Reyle. 1993. From discourse to logic - introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory. In *Studies in Linguistics and Philosophy*.

Rik van Noord, Lasha Abzianidze, Antonio Toral, and Johan Bos. 2018. Exploring neural methods for parsing discourse representation structures. *Transactions of the Association for Computational Linguistics*, 6:619–633.

Rik van Noord, Antonio Toral, and Johan Bos. 2020. Character-level representations improve DRS-based semantic parsing even in the age of BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4587–4603, Online. Association for Computational Linguistics.

Geoffrey Nunberg, Ivan A. Sag, and Thomas Wasow. 1994. Idioms. *Language*, 70(3):491–538. Publisher: Linguistic Society of America.

Jan Odijk and Martin Kroon. 2024. A canonical form for flexible multiword expressions. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 91–101, Torino, Italia. ELRA and ICCL.

Carlos Ramisch, Silvio Ricardo Cordeiro, Agata Savary, Veronika Vincze, Verginica Barbu Mititelu, Archna Bhatia, Maja Buljan, Marie Candito, Polona Gantar, Voula Giouli, Tunga Güngör, Abdelati Hawwari, Uxoa Iñurrieta, Jolanta Kovalevskaitė, Simon Krek, Timm Lichte, Chaya Liebeskind, Johanna Monti, Carla Parra Escartín, Behrang QasemiZadeh, Renata Ramisch, Nathan Schneider, Ivelina Stoyanova, Ashwini Vaidya, and Abigail Walsh. 2018. Edition 1.1 of the PARSEME shared task on automatic identification of verbal multiword expressions. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 222–240, Santa Fe, New Mexico, USA. 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.

Chunliu Wang, Huiyuan Lai, Malvina Nissim, and Johan Bos. 2023. Pre-trained language-meaning models for multilingual parsing and generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5586–5600, Toronto, Canada. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Xiao Zhang, Chunliu Wang, Rik van Noord, and Johan Bos. 2024. Gaining more insight into neural semantic parsing with challenging benchmarks. In *Proceedings of the Fifth International Workshop on Designing Meaning Representations @ LREC-COLING 2024*, pages 162–175, Torino, Italia. ELRA and ICCL.

Does discourse structure help action prediction? A look at Correction Triangles

Kate Thompson[†‡], **Akshay Chaturvedi**[†‡], **Nicholas Asher***[‡]
[†]LINAGORA Labs, [‡]IRIT, *CNRS
Toulouse, France

Abstract

An understanding of natural language corrections is essential for artificial agents that are meant to collaborate and converse with humans. We present some preliminary experiments using language-to-action models investigating whether discourse structure, in particular CORRECTION relations, improves the action prediction capabilities of language-to-action models for simple block world tasks. We focus on scenarios in which a model must correct a previous action, and present a corpus of synthetic dialogues to help explain model performance.

1 Introduction

In order to successfully complete a shared task, such as building a block structure in a shared environment, participants must accumulate a body of shared information about the goal of the task, the changing state of play, and their beliefs and intentions, collectively referred to as *common ground* (Clark, 1996). Errors are integral to the process of building common ground, as participants naturally explore and test their strategies through trial and error (Thomaz et al., 2019). When natural language is among the modes of communication available to participants, corrective speech acts provide an efficient and information-rich mechanism with which they can identify and quickly resolve errors (Benotti and Blackburn, 2021). For artificial agents that can collaborate with humans using natural language, the ability to understand and use corrections is essential.

While most speech acts entail a monotonic update of the common ground (elaborations or acknowledgments, for example), a correction entails a revision to the common ground, and is an example of a *divergent* speech act (Asher and Lascarides, 2003). An agent that understands corrections must:

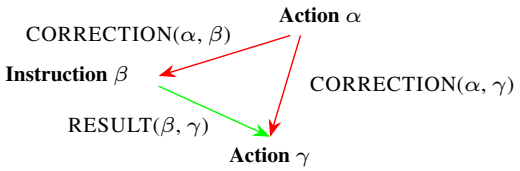


Figure 1: A **Correction Triangle** is formed by two CORRECTION relations and a RESULT relation, and appears in dialogues where an initial correction to an action results in a new action.

1. Recognize an utterance as an instance of a divergent speech act.

2. Determine the content of the correction by identifying the parts of the previous dialogue and/or shared environment it refers to.

3. Revise the common ground according to 1 and 2, and make any required changes to the shared environment.

The Minecraft Dialogue Corpus (MDC) (Narayan-Chen et al., 2019) features interactions in which two humans, playing the roles of *Builder* and *Architect*, collaborate to construct block structures in a simulated 3D environment. Architect and Builder communicate via chat window, where Architect describes the structure to Builder, who may then place and remove blocks on the grid. The MDC provides paradigmatic examples of collaborative conversation situated in a shared environment, wherein the players' linguistic contributions and the non-linguistic Builder actions are highly interdependent, particularly when Builder performs an erroneous action which the Architect must then verbally correct. The Minecraft Structured Dialogue Corpus (MSDC) (Thompson et al., 2024b) adds an annotation layer to the MDC, drawing semantically-typed relations (Asher and Lascarides, 2003) between player utter-

Proceedings of the 16th International Conference on Computational Semantics, pages 176–184
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

ances and actions, which elucidate the overarching semantic structure—or *discourse structure*—of the interaction. In the MSDC, correction scenarios are captured in substructures called *Correction Triangles*: an Architect instruction β stands in the CORRECTION relation to the previous erroneous action α, eliciting a new action γ, which is the RESULT of the Architect correction and also a CORRECTION of the erroneous action (Figure 1).

The MSDC discourse structures, including Correction Triangles, were shown to be automatically retrievable with state-of-the-art accuracy using the Llamipa discourse parser (Thompson et al., 2024a). Presumably, an agent utilizing the output of this parser would have at least a partial understanding of correction scenarios, since predicting the relation CORRECTION(α, β), amounts to identifying utterance α as a correction (*"no, I said add a blue block"* (1), and connecting it to the previous context β (*Builder places red block*) (2). However, the question we want to address in this paper is whether an agent can leverage the discourse structure of an interaction to inform its subsequent manipulation of the environment. This goes beyond the parsing task: given a dialogue context and its semantic structure, including CORRECTION(α, β), we want to know whether the presence of the structure improves agent predictions of the action sequence γ that would appropriately complete the Triangle.

In what follows, we briefly discuss current work relevant to our question. We then describe the language-to-action model that will serve as our agent for action prediction experiments using the MSDC. We explain how our preliminary results incentivize the creation of synthetic correction dialogues, which allow for more tightly controlled experiments. We then discuss model performance on synthetic correction dialogues and directions for future work.

2 Related Work

Previous approaches to building agents that understand corrections in collaborative tasks differ with respect to model architectures. Rubavicius et al. (2024) and Appelgren and Lascarides (2020) build agent models based on pared down cognitive architectures for interactive task learning (Laird et al., 2017), where a corrective utterance generates symbolically encoded, probabilistically weighted hypotheses, and is used to update the agent's belief state and inform an action plan. Alternatively, Chiyah-Garcia et al. (2024) create a language-to-action model by fine-tuning a large vision language model (VLM) on instruction-action pairs from a block world dataset (Bisk et al., 2016). They augment the instructions with third position repairs (Schegloff, 1992), and use masking techniques during finetuning to encourage the model to recognize the repair. In our experimental setup, we also use an LLM-based language-to-action model: our agent model is based on Nebula (Chaturvedi et al., 2024), a Llama3 architecture fine-tuned on the MDC dialogue-to-action corpus. Furthermore, we use the discourse structure annotations from the MSDC, which make corrections explicit, whereas the repairs in Chiyah-Garcia et al. (2024) were unmarked.

Discourse parsing predicts the semantic relations that hold between the elementary units of a dialogue, and produces a structural representation of the interaction. The most recent approaches to parsing based on LLMs formulate the structure prediction task as a sequence-to-sequence generation task (Li et al., 2024), where the parser takes the dialogue units as input, and outputs the discourse structure as a sequence of typed tuples. The Llamipa parser (Thompson et al., 2024b) provides state-of-the-art results on the MSDC using this approach. Our agent model uses the Llamipa structure representation, in which the discourse graph is flattened into a string of typed tuples, where each tuple represents a single relation (see Figure 2).

Discourse structure has been used to improve performance on various downstream tasks. Devatine et al. (2023) leverages discourse information to predict political orientation of news articles, while Rennard et al. (2024) uses it to improve extractive meeting summarization. Sharma et al. (2025) demonstrate that discourse structure can improve a model's performance on mathematical reasoning tasks. The experiments described in this paper are the first to use discourse structure to improve action prediction in situated collaborative tasks.

Finally, data synthesis has been increasingly used to provide high-quality training data for LLMs and LLM-based agents (Liu et al., 2024; Shichman et al., 2024), as well as to perform targeted tests of LLM knowledge (Wu et al., 2024). Synthetic data has been shown to be especially helpful in determining which concepts an agent trained on the MDC's ambiguous natural language instructions actually learned (Chaturvedi et al., 2024; Jayan-

	MDC (F1)		SynthCorr300 (Accuracy) [Action error/ Site error]			
	All relations	Correction	Overall	D1	D3	D5
Nebula	**0.39**	**0.57**	0.79 [0.18/0.03]	**1.00** [0/0]	0.73 [0.07/0.02]	0.64 [0.1/0.02]
Nebulipa	0.37	0.50	**0.80** [0.17/0.02]	**1.00** [0/0]	**0.75** [0.06/0.02]	**0.66** [0.1/0.01]
Nebulipa-E	0.37	0.52	0.67 [0.29/0.04]	0.98 [0.01/0]	0.57 [0.12/0.02]	0.46 [0.17/0.01]

Table 1: Performance of the context-aware (Nebula) and structure-aware (Nebulipa) models. Nebulipa-E(mpty) shows the results of an ablation in which structure is removed from the test samples. Column 1 shows the net action F1 scores on the MDC **test** set using all relation types; Column 2 shows F1 calculated on just those MDC **validation** samples whose predicted action sequence is mediated by a CORRECTION relation (see Figure 3). For a full breakdown of MDC splits see Appendix C. Columns 3-6 give the accuracy scores on the 300 synthetic correction dialogues broken down by the CORRECTION distance.

navar et al., 2025). This work is the first to create synthetic dialogues with discourse annotations to test the efficacy of discourse structure in a downstream task.

3 A structure-aware language-to-action model

The Nebula language-to-action model (Chaturvedi et al., 2024), was trained using the MDC to predict a Builder action sequence given the previous dialogue, and was evaluated using the same net action F1 metric as the baseline MDC model (Jayannavar et al., 2020). Given a completed action sequence, net action F1 is computed on newly placed blocks that exactly match the color and position of those in the corresponding gold action sequence. Nebula leveraged the large context window of the Llama3-8b architecture (Dubey et al., 2024), which allowed it to predict an action sequence using the entire previous dialogue context, resulting in a *context-aware* model that doubled the baseline F1 on the MDC.

In order to see whether the addition of discourse structure might further improve performance on the action prediction task, we augmented each MDC training sample with the gold[1] discourse structure from the MSDC. We formatted the structure as a sequence of typed tuples, and appended it to the dialogue context (see Figure 2). Following the Nebula training regime, we finetuned Llama3-8b using QLoRA (Dettmers et al., 2023) for 3 epochs on the augmented data (training parameters shown in Appendix A). The result was *Nebulipa*, a *structure-aware* language-to-action model.

The leftmost column of Table 1 compares the net action F1 scores of Nebula and Nebulipa on the MDC augmented with the full MSDC structures; i.e., all 17 relation types (Thompson et al., 2024b). Nebulipa-E ("Nebulipa-Empty") shows the result of an ablation in which the structure was removed from the test samples, in order to provide some further indication of whether Nebulipa, trained with structure, learned to use it. We see that this brute inclusion of structure hinders rather than improves model performance, as F1 drops two points. Also, Nebulipa-E shows no change with respect to Nebulipa. If Nebulipa were using discourse information for action prediction, we would expect its removal to result in a drop in F1; yet this result indicates that, overall, training with structure did not lead to the model to exploit it.

4 Focusing on Correction Triangles

The preliminary result above shows that including full discourse structures, containing many different relation types, does not improve language-to-action model performance on the MDC action prediction task. We note that the relational structure presents each relation uniformly, even though some types are more informative than others, given the discourse context. As mentioned in Section 1, CORRECTIONS describe revisions to the common ground, and so are often more informative than other relation types holding between less salient parts of the context: COMMENT, ACKNOWLEDGEMENT, etc.

To test this, we took a subset of MDC samples[2] in which the final action sequence to be predicted by the model is the result of an Architect correction, i.e. where a CORRECTION relation is critical to the final prediction (Figure 3). The second column of

[1] Since the purpose of these experiments is to see whether the inclusion of structure makes any difference at all, we use the gold annotations. Of course, a fully autonomous agent would predict actions as well as structure.

[2] For this test we looked at a subset of the MDC validation set, 149 of 1051 samples. See Appendix C for a description of MDC splits.

Table 1 shows that F1 improves on the subset, but is still higher for Nebula, thus corroborating the first result. Further, F1 *improves* when structure is removed (Nebulipa-E), suggesting that the addition of structure hinders performance.

Nevertheless, we maintain that this result must not be taken as decisive for three reasons. The first is that the longer dialogue contexts feature a dense relational structure (with uniform relations, as just mentioned), presenting the possibility that the signal provided by more informative relation types, such as CORRECTION, is greatly diminished; this is illustrated in Appendix B. Second, the MDC instructions contain highly context-dependent language, rich in anaphora and ellipsis, which often does not indicate a single correct action sequence. For example, we see in Figure 3 that Nebulipa performs the correct net action given the previous dialogue, but then places additional blocks—yet there is nothing in the dialogue that prohibits this. The second reason, already mentioned above, is that the contexts are long and can be very dense. Lastly, since the net action F1 metric requires action sequences match the exact positions of the gold sequences, it unjustly discounts actions that are the result of instructions that are naturally ambiguous, e.g. *put a block in a corner*, and thus obscures the model's true performance.

Taking these factors into consideration, we determined that a set of short dialogues in simple correction scenarios, in which we could zero in on Correction Triangles, would help us more clearly assess whether structure can be leveraged for action prediction. To this end, we synthetically generated *SynthCorr300*, a set of 300 short dialogues[3]. In each dialogue, Architect gives three instructions for simple shapes, one of which Builder botches, eliciting a correction in Architect's final turn (Figure 2). The shapes are towers and rows, which Nebula was shown to build with high accuracy (Chaturvedi et al., 2024), as well as single blocks, which Nebula was able to place and remove on rows and towers already present on the grid. For each instruction, a shape and its parameters were chosen randomly: one of six possible colors and a size (for towers and rows) of 3, 4, or 5 blocks. Repeated colors, shapes, or sizes occurs in a majority of the dialogues (Table 2), however, each shape is disambiguated by its location descriptor (*centre*, *corner edge*). To botch an instruction for a tower or row, we randomly chose whether to remove or add (+1 or -1) a single block from the number of blocks given by Architect. For single block placement instructions, we changed the color of the block given by Architect by randomly choosing from the five remaining colors.

0 <Build> Mission has started.
1 <Arch> Let's start with some basic shapes.
2 <Arch> Build a row of 3 red blocks at the centre.
3 <Build> `place red 010, place red 110`
4 <Arch> And place a red block at an edge.
5 <Build> `place red -215`
6 <Arch> Also build a yellow tower of size 3 at a corner.
7 <Build> `place yellow -515, place yellow -525, place yellow -535`
8 <Arch> The red row at the centre should be 3 blocks.

STRUCTURE:
Continuation(0,1), Continuation(1,2), Result(2,3), Continuation(2,4), Result(4,5), Continuation(4,6), Result(6,7), Correction(3,8)

BUILDER PREDICTION:
`place red -110 [OR place red 210]`

Figure 2: A SynthCorr300 dialogue example in which the given CORRECTION connects the Architect at turn 8 with the Builder error at turn 3, and so is of distance 5 (D5). NB: the SynthCorr300 dialogues use the Llamipa structure representation (Thompson et al., 2024a), where the relations are typed tuples appended after the dialogue in a "Structure" field.

We generated 100 dialogues for each of the three possibilities for Builder error: after the first, second, or third instruction. We also generated the discourse structure, which was identical for each dialogue except for the first relation of the Correction Triangle CORRECTION(α, β). This latter varied with the position of the error, e.g. if it was after the first instruction, the CORRECTION would reach farther back into the dialogue context (distance 5) than it would if it was after the second (distance 3) or third (distance 1) instruction.

SynthCorr300 tests whether a model can accurately produce the action sequence γ which would effectively complete the Correction Triangle, and whether the addition of discourse structure improves its performance. The correct action sequence for each dialogue is clearly defined and can be checked automatically.

[3]The SynthCorr300 data, and the code used to generate it, are available at https://huggingface.co/datasets/linagora/synthetic_corrections

55. Arch: ok
56. Arch: two more orange blocks to go
57. Arch: these are on the ground
58. Arch: on the diagonal back toward the line we started on
59. Buil: place orange -3 1 1, place orange -2 1 0
60. Arch: almost
61. Arch: but the other way
62. Arch: and shifted one block away from the structure
63. Buil: pick -3 1 1, pick -2 1 0, place orange -3 1 -1, place orange -2 1 0
64. Buil: here?
65. Arch: not quite
66. Buil: grr, okay

MDC Gold:	pick -2 1 0, pick -3 1 -1
Nebula:	pick -3 1 -1, pick -2 1 0
Nebulipa:	pick -2 1 0, pick -3 1 -1, place orange -2 1 -1, place orange -1 1 -1
Nebula-E:	pick -2 1 0, place orange -3 1 0, place orange -2 1 -1, pick -3 1 0

Figure 3: MDC sample where the predicted sequence is the RESULT of Architect CORRECTION in 65 and a CORRECTION of the action sequence in 63.
NB: to conserve space we truncated the dialogue context of this sample, shown in full in Appendix B.

	D1	D3	D5	All
No ambiguity	17	12	13	42
Color only	7	11	9	27
Shape only	41	39	43	123
Color and shape	35	38	35	108

Table 2: Sample counts by relation distance D between the CORRECTION source and target, and ambiguity type: *Color only*: shapes are all different, but colors are repeated; *Shape only*: colors are different but shapes are repeated.

5 Results

There was only a one-point difference between Nebula and Nebulipa on SynthCorr300 (Table 1), but unlike on MDC tests, Nebulipa was in the superior position. Furthermore, Nebulipa-E ablation conformed to previous expectations: accuracy dropped substantially when the structure was removed, indicating that a model trained with structure does learn to leverage it. When we look at the performance by CORRECTION distance, we see that the pattern holds at longer distances, although performance degraded for all models as distance increased, which is unsurprising given that the majority of CORRECTIONS in the MSDC training data ($\sim$ 70%) are of distance 3 or less.

To consider another angle of comparison, we looked at two ways in which models failed to generate the correct action sequences. *Action errors* occurred when the model correctly identified the shape to be changed (after the first, second, or third instruction), but did not perform the correct actions to do so. *Site errors* occurred when the model changed a shape that was not indicated by the CORRECTION, misidentifying the botched sequence. Returning to the discussion in Section 1 of what is in involved in understanding corrections, we can roughly align Site errors with the failure to identify what portion the previous dialogue the CORRECTION refers to (2), e.g. which instruction. We can align Action errors with the failure to properly revise the common ground (3), e.g. to perform the correct block placements and removals.

Table 1 gives action errors and site errors as a proportion of *total* samples. There was little difference in error rates between Nebula and Nebulipa, although with Nebulipa-E we saw an increase in Action errors. Since a CORRECTION representation *Corr(x,y)* effectively acts as a pointer to the botched sequence x, we would expect an increase in Site errors once the pointer was taken away. Instead, there was a greater incidence of Action errors. While the SynthCorr300 data is too small to support conclusions on the relationship between error types, the preliminary indication here is that the CORRECTION relations are not used to pick out the error site (perhaps the model can already do this using linguistic context) but rather to provide important semantic information about what the model is supposed to do at the site. For instance, it is possible the Architect utterance *"The tower ... should be 3 blocks."* might only lead to model to correctly infer the correct actions (i.e., *remove one block from the tower*) when combined with a semantic marker for CORRECTION. Possible future work might thus involve varying the synthetic dialogues by replacing the final CORRECTIONS with a different but coherent relation type such as ELABORATION, and testing for changes in Action errors.

6 Conclusion

In this paper we broach a question in discourse that has gotten relatively little attention—*can discourse structure guide action predictions?*—and explore a particular LLM-based approach for an initial investigation. The results of our synthetic dialogue experiments showed that, overall, access to large contexts overrides the effects of adding explicit discourse representations. However, there was some indication that models trained with structure did learn to exploit CORRECTIONS, using them to correct relevant parts of the discourse context with higher accuracy. In future work, we will enlarge the synthetic data in order to further investigate the action error results, as well as consider more varied correction scenarios.

7 Limitations

This work explores the role of discourse structure in conversational instruction following scenarios where the agent's goal is to perform the correct action sequences in a shared environment. It only considers one instantiation of such a scenario, in which the agent builds block structures on a 3D grid. The experimental results are obtained using agent models based on generative LLMs, where the discourse structure is represented as a string of typed tuples, and appended after the dialogue text in the model inputs. Certainly there are other ways to represent and feed structure into the agent model—or perhaps to integrate structural information into the model architecture rather than the data inputs—as well as other model architectures which would be worth exploring, such as graph neural networks. The synthetic data generated is small, and covers only a portion of the variation possible in correction dialogues with respect to the source and complexity of the Builder action errors, and to the referential ambiguity of correction language.

Acknowledgments

We thank our reviewers for their observations and constructive criticism. For financial support, we thank the National Interdisciplinary Artificial Intelligence Institute ANITI (Artificial and Natural Intelligence Toulouse Institute), funded by the French 'Investing for the Future–PIA3' program under the Grant agreement ANR-19-PI3A-000. This project has also been funded by the France 2030 program and is funded by the European Union - Next Generation EU as part of the France Relance. We also thank the projects COCOBOTS (ANR-21-FAI2-0005) and DISCUTER (ANR-21-ASIA-0005), and the COCOPIL "Graine" project of the Région Occitanie of France. This work was granted access to the HPC resources of CALMIP supercomputing center under the allocation 2016-P23060.

References

Mattias Appelgren and Alex Lascarides. 2020. Interactive task learning via embodied corrective feedback. *Autonomous Agents and Multi-Agent Systems*, 34:1–45.

Nicholas Asher and Alex Lascarides. 2003. *Logics of Conversation*. Cambridge University Press.

Luciana Benotti and Patrick Blackburn. 2021. Grounding as a collaborative process. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 515–531, Online. Association for Computational Linguistics.

Yonatan Bisk, Deniz Yuret, and Daniel Marcu. 2016. Natural language communication with robots. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 751–761.

Akshay Chaturvedi, Kate Thompson, and Nicholas Asher. 2024. Nebula: A discourse aware Minecraft builder. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6431–6443, Miami, Florida, USA. Association for Computational Linguistics.

Javier Chiyah-Garcia, Alessandro Suglia, and Arash Eshghi. 2024. Repairs in a block world: A new benchmark for handling user corrections with multi-modal language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11523–11542, Miami, Florida, USA. Association for Computational Linguistics.

Herbert H Clark. 1996. *Using language*. Cambridge university press.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.

Nicolas Devatine, Philippe Muller, and Chloé Braud. 2023. An integrated approach for political bias prediction and explanation based on discursive structure. In *Findings of the Association for Computational Linguistics (EACL 2023)*, pages 11196–11211. ACL: 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, et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Prashant Jayannavar, Anjali Narayan-Chen, and Julia Hockenmaier. 2020. Learning to execute instructions in a Minecraft dialogue. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2589–2602, Online. Association for Computational Linguistics.

Prashant Jayannavar, Liliang Ren, Marisa Hudspeth, Charlotte Lambert, Ariel Cordes, Elizabeth Kaplan, Anjali Narayan-Chen, and Julia Hockenmaier. 2025. Bap v2: An enhanced task framework for instruction following in minecraft dialogues. *arXiv preprint arXiv:2501.10836*.

John E Laird, Kevin Gluck, John Anderson, Kenneth D Forbus, Odest Chadwicke Jenkins, Christian Lebiere, Dario Salvucci, Matthias Scheutz, Andrea Thomaz,

Greg Trafton, et al. 2017. Interactive task learning. *IEEE Intelligent Systems*, 32(4):6–21.

Chuyuan Li, Yuwei Yin, and Giuseppe Carenini. 2024. Dialogue discourse parsing as generation: A sequence-to-sequence LLM-based approach. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–14, Kyoto, Japan. Association for Computational Linguistics.

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, et al. 2024. Best practices and lessons learned on synthetic data. *arXiv preprint arXiv:2404.07503*.

Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. 2019. Collaborative dialogue in Minecraft. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5405–5415, Florence, Italy. Association for Computational Linguistics.

Virgile Rennard, Guokan Shang, Michalis Vazirgiannis, and Julie Hunter. 2024. Leveraging discourse structure for extractive meeting summarization. *arXiv preprint arXiv:2405.11055*.

Rimvydas Rubavicius, Peter David Fagan, Alex Lascarides, and Subramanian Ramamoorthy. 2024. Secure: Semantics-aware embodied conversation under unawareness for lifelong robot learning. *arXiv preprint arXiv:2409.17755*.

Emanuel A Schegloff. 1992. Repair after next turn: The last structurally provided defense of intersubjectivity in conversation. *American journal of sociology*, 97(5):1295–1345.

Krish Sharma, Niyar R Barman, Akshay Chaturvedi, and Nicholas Asher. 2025. Dimsum: Discourse in mathematical reasoning as a supervision module.

Mollie Frances Shichman, Claire Bonial, Taylor A. Hudson, Austin Blodgett, Francis Ferraro, and Rachel Rudinger. 2024. PropBank-powered data creation: Utilizing sense-role labelling to generate disaster scenario data. In *Proceedings of the Fifth International Workshop on Designing Meaning Representations @ LREC-COLING 2024*, pages 1–10, Torino, Italia. ELRA and ICCL.

Andrea L Thomaz, Elena Lieven, Maya Cakmak, Joyce Y Chai, Simon Garrod, Wayne D Gray, Stephen C Levinson, Ana Paiva, and Nele Russwinkel. 2019. Interaction for task instruction and learning. In *Interactive task learning: Humans, robots, and agents acquiring new tasks through natural interactions*, pages 91–110. MIT Press.

Kate Thompson, Akshay Chaturvedi, Julie Hunter, and Nicholas Asher. 2024a. Llamipa: An incremental discourse parser. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6418–6430, Miami, Florida, USA. Association for Computational Linguistics.

Kate Thompson, Julie Hunter, and Nicholas Asher. 2024b. Discourse structure for the Minecraft corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4957–4967, Torino, Italia. ELRA and ICCL.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. 2024. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. 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 1819–1862, Mexico City, Mexico. Association for Computational Linguistics.

A Model Parameters

GPUs	
4 NVIDIA Volta V100	
Hyperparameters	
Training epochs	3
batch size	4
optimizer	Adam
learning rate	2e-4
learning rate scheduler	linear warm-up and cosine annealing
warm-up ratio	0.03
gradient clipping	0.3
lora r	64
lora (alpha)	16
lora dropout ratio	0.1
lora target modules	Only Attention Blocks (q_proj, v_proj)
quantization for Llama3	4-bit NormalFloat

Table 3: Details on computing resources and hyperparameters for finetuning Llamipa.

Table 3 gives the hyperparameters used for finetuning Nebula and Nebulipa, along with the computing resources. We adapt the finetuning code from the following repository[4].

[4] https://github.com/mlabonne/llm-course/blob/main/Fine_tune_Llama_2_in_Google_Colab.ipynb

B MDC sample

Figure 4: MDC sample where the predicted sequence is the RESULT of Architect CORRECTION in 65 and a CORRECTION of the action sequence in 63. The full discourse structure is given with CORRECTIONS highlighted. The Correction Triangles are superimposed on the context for reference.

C MDC data

	MDC test		MDC validation			
	# samples	F1	# samples	F1	# Correction samples	F1
Nebula	1615	0.39	1335	0.39	149	0.57
Nebulipa	1471	0.37	1194	0.355	149	0.50
Nebulipa-E	1471	0.37	1194	0.363	149	0.52

Table 4: Number of samples and model F1 for the MDC test and validation splits. The Correction set discussed in Section 4 is a subset of the validation set.

The Nebula language-to-action model predicts Builder actions given the entire previous dialogue context. We prepared the MDC dialogues for training and testing Nebula by dividing it into dialogue-action pairs. Thus the number of data samples in a split is equal to the total number of Builder actions across all dialogues in that split. The 100 validation dialogues contain 1335 action sequences, and the 101 test dialogues contain 1615 action sequences—the dialogues in test were on average longer (had greater number of utterances) than those in validation (Thompson et al., 2024b).

The MSDC [5] provides complete dialogue structure annotations for all MDC dialogues. Adding structure from the MSDC to the MDC samples for Nebulipa training was not straightforward. During the MSDC annotation campaign, some of the Builder action sequences were fused together into *Complex Discourse Units* (see Figure 1 in Thompson et al. (2024b)), which lead to a reduction in the overall number of separate action sequences, as can be seen in Table 4. When the action sequences were combined, the dialogue moves between them also shifted. As a result, the MDC data used for Nebulipa is not *identical* to the data used for Nebula (ignoring the addition of structure to the Nebulipa data). However, the overlap between them is large enough to warrant their comparison: 116 of the 1194 ($\sim$ 9%) samples in the Nebulipa validation set were not present in the Nebula set, and 99 out of 1471 ($\sim$ 7%) of samples in the test set.

In Section 4 we isolated the samples in the validation set where the action sequence to be predicted was the target of a CORRECTION relation. There were 182 such samples in the Nebulipa data, but only 149 of these were also present in the Nebula data.

[5] https://huggingface.co/datasets/linagora/MinecraftStructuredDialogueCorpus

FAMWA: A new taxonomy for classifying word associations (which humans improve at but LLMs still struggle with)

Maria A. Rodriguez[1,2], **Marie Candito**[3],
Richard Huyghe[1]
[1] University of Fribourg,
[2] Lucerne University of Applied Sciences and Arts,
[3] LLF (Université Paris Cité / CNRS)

Abstract

Word associations have a longstanding tradition of being instrumental for investigating the organization of the mental lexicon. Despite their wide application in psychology and psycholinguistics, analyzing word associations remains challenging due to their inherent heterogeneity and variability, shaped by linguistic and extralinguistic factors. Existing word-association taxonomies often suffer limitations due to a lack of comprehensive frameworks that capture their complexity. To address these limitations, we introduce a linguistically motivated taxonomy consisting of co-existing meaning-related and form-related relations, while accounting for the directionality of word associations. We applied this taxonomy to a dataset of 1,300 word associations (FAMWA) and assessed it using various LLMs, analyzing their ability to classify word associations. The results indicate higher inter-annotator agreement with our taxonomy compared to previous studies (κ = .60 for meaning and κ = .58 for form). However, models such as GPT-4o perform only modestly in relation labeling (with accuracies of 46.2% for meaning and 78.3% for form), which calls into question their ability to fully grasp the underlying principles of human word associations.

1 Introduction

The word association task is a classic psychological experiment in which participants respond spontaneously with the first word(s) that come to mind (e.g., *cat*, *bark*, *bone*) when presented with a specific cue word (e.g., *dog*). For more than a century, word associations have been used by psychologists and psychiatrists to investigate cognitive processes, psychological behavior patterns, mental disorders, language acquisition, multilingualism, and the overall structure of the mental lexicon (see Galton 1879; Jung 1910; Kent and Rosanoff 1910; Deese 1965; Riegel and Zivian 1972; Meara 1983; a.o.). Experiments conducted across multiple languages have shown that word associations are characterized by both heterogeneity and variability. On the one hand, responses may be influenced by a wide range of relationships between cues and responses, depending on formal, semantic, and syntactic properties, as well as extralinguistic knowledge and cultural factors. On the other hand, there is considerable variation in responses across individuals, with greatly varying degrees of convergence depending on the cue word. This diversity poses challenges for linguistic analysis, and various taxonomies of word-association relations have been proposed to accurately account for word associations.

Although common elements of classification emerge from previous studies, there is no universally accepted framework for describing word-association relations, as existing taxonomies present various limitations. The categories used to analyze associations are not always explicitly defined, and some taxonomies focus only on a single aspect of word-association relations (e.g., semantic characteristics), while others merge semantic and formal relations into flat hierarchies, potentially leading to conflicting or incomplete descriptions. In addition, directionality is rarely considered, although the cue-response sequencing and the non-reciprocal nature of word associations call for a specific account of asymmetrical relations. Finally, taxonomies are seldom evaluated through inter-annotator agreement or computational models, limiting their validation and reliability.

In this study, we propose a linguistically motivated taxonomy of word-association relations based on a critical examination of previous classifications. The taxonomy includes two co-existing levels of linguistic analysis related to form and meaning, and takes into account the directionality of word associations. We apply this taxonomy

Proceedings of the 16th International Conference on Computational Semantics, pages 185–198
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

on a dataset of 1,300 word associations in English using a well-defined annotation protocol, while assessing annotation quality through inter-annotator agreement. Furthermore, we evaluate the ability of generative language models to classify word associations according to our taxonomy, exploring how well they capture the diversity of relations in word associations, and providing a detailed analysis of model performance. Overall, the main contributions of the study include (i) a theoretical and methodological reflection on the analysis of word-association relations, (ii) the creation of a finely annotated dataset of word associations covering formal and semantic relations, and (iii) a discussion of how language models handle the heterogeneity of lexical relations within word associations.

2 Background

2.1 Linguistic description of word associations

A close examination of word associations reveals that they are not restricted to lexical relations, which alone cannot account for the full range of relationships observed between cues and responses (see, e.g., Schulte im Walde et al. (2008) for German). Across the literature, linguistic descriptions addressing specifically word associations often distinguish between syntagmatic, paradigmatic, and clang associations (see, e.g., Deese 1962; Glanzer 1962). Syntagmatic associations are observed between words that may cooccur in context (e.g., *friend-best*), whereas paradigmatic associations involve words from the same lexical class with related meanings (e.g., *certain-sure*), and clang associations are based on phonological similarities between cues and responses (e.g., *hat-cat*). Traditionally, these categories have been considered as mutually exclusive, which presents challenges when analyzing word pairs that could belong to multiple categories. More broadly, the tripartite classification between syntagmatic, paradigmatic, and clang associations has been criticized for being too coarse-grained, while still failing to account for all word associations, and relying on overly vague category definitions.

To address these challenges, more fine-grained analyses of word-association relations have been proposed. Fitzpatrick (2006, 2007) introduced a taxonomy based on 4 main categories (meaning-based, position-based, form-based, and erratic), further divided into 17 subcategories for a more detailed classification. Similarly, Santos et al. (2011) used 10 basic categories to describe response words, notably accounting for the directionality of associations (Tversky, 1977). For instance, they distinguished between "domain higher category" and "domain lower category" to differentiate cases in which the response represents a superordinate or a subordinate concept relative to the cue. However, these fine-grained taxonomies still conflate the formal and semantic aspects of word-association relations into a single classification, implying a complementary distribution that does not always apply in practice.

Another classification approach is based on the system proposed by Wu and Barsalou (2009), originally designed to analyze concept representations, but later applied to semantic feature norming (Bolognesi et al., 2017; Vivas et al., 2022) and word associations (Liu et al., 2022; De Deyne et al., 2024). This framework distinguishes between taxonomic, situational, entity, and introspective properties, with the potential for further division into more detailed classes (see, e.g., McRae et al. 2012). However, an inherent limitation of this taxonomy is that not all relations in word associations are based on property descriptions, nor are they always determined semantically. As a result, restricting the analysis to this taxonomy may lead to an incomplete characterization of word associations, particularly by overlooking their more formal aspects.

Three key observations emerge from the discussion above. First, both formal (i.e., morphological and phonological) and semantic relations can drive word associations, and a comprehensive taxonomy should integrate both aspects to fully capture word-association relations. Second, while formal and semantic relations should be distinguished, they should not be treated as mutually exclusive, as there is no logical incompatibility in formal and semantic motivations for word associations. A multilevel analysis is necessary to reflect both the linguistic relations underlying word associations and the complexity of the cognitive processes involved. Third, a detailed analysis of word-association relations must account for both symmetrical (e.g., synonymy, phonological resemblance) and asymmetrical relations (e.g., hyponymy, morphological derivation). Given that word associations are by definition oriented from cues to responses, and reciprocity between them is rarely observed, taxonomies including directional classes are essential to provide a fine description of word associations.

2.2 Existing datasets

Word associations have been collected for various languages, on a growing scale over the years (see, e.g., Kiss et al. 1973; Nelson et al. 2004 for English). SWOW is currently the largest multilingual word-association dataset, covering 19 languages[1]. In this paper, we focus on its English part, which was collected via crowdsourcing (De Deyne et al., 2019). For each of 12,282 English cues, 100 participants were asked to answer the first 3 words coming to their mind, resulting in a dataset of over 150k unique cue-response pairs, each associated with the number of participants who answered the response at each position (hereafter R1, R2, and R3). De Deyne et al. (2019) checked that the continued response paradigm of the English SWOW and the more heterogeneous participant sample did not affect properties compared to other single-response English datasets, and observed small evidence for response chaining—cases of R2 being influenced by R1 response.

Smaller word association datasets include the labeling of cue-response pairs into categories, possibly based on participants' explanations for the associations. For example, Fitzpatrick (2006) collected single-response associations from 40 participants for 60 cues, conducted retrospective interviews, and categorized the associations according to the participants' explanations. Similarly, Liu et al. (2022) asked participants to both produce associations and explain their responses, compiling the WAX dataset, which contains 15k unique cue-response pairs and 19k cue-response-explanation triples. Among these, 1,602 triples were classified into 16 word-association categories, half manually by humans and half automatically, based on the identification of explanation patterns associated with certain labels—a method that may affect the reliability of the classification. The inter-annotator agreement for the human classification was measured but found to be only moderate (Cohen's κ = .42). A possible limitation of the explanation-based approach is that, although prompting participants to provide explanations for their associations may help clarify the cue-response relation, it can also introduce bias by making responses less spontaneous, as already observed by Woodworth (1938).

[1] https://smallworldofwords.org/en/project/home

2.3 Computational approaches to word associations

Studies on word associations using pre-trained language models have developed recently, following three lines of research. Some researchers have compared the properties of word associations with those of word embeddings. For instance, A. Rodriguez and Merlo (2020) found that the top-K neighbors of a cue encoded with BERT (Devlin et al., 2019) often contain human responses.

Computational models have also been employed to mimick the word association task. Vintar et al. (2024) prompted encoder-decoder language models to provide an unlimited list of response words given a cue in Slovene and English, from the English and Slovene SWOW datasets. Abramski et al. (2025) prompted three decoder-only large language models (LLMs) to produce three word associations, using the English SWOW cues. While both works report relatively low overlap between the human and models' associations, Abramski et al. (2025) found that human and models' responses do share semantic properties: when building a semantic network based on the associations (one network for human associations, and one network per prompted LLM), the authors report the same strong correlation level between the ease of lexical retrieval for human participants and the closeness in the semantic word association network[2], for all the four networks (one human, and three LLM-based).

A third line of research focuses on learning or using models to classify cue-response relations. For example, Liu et al. (2022) used the WAX dataset, which contains cue-response-explanation triples, and designed various tasks to assess how well language models capture the underlying relations between cues and responses. In particular, they trained relation classifiers based on BERT and BART (Lewis et al., 2020), but reported relatively low performance (weighted F1 = 48%). Similarly, De Deyne et al. (2024) prompted GPT-4 to classify a fraction of the human-labeled part of the WAX dataset (among other datasets[3]). They reported a classification F-score of 47%, indicating that the

[2] There is a negative correlation between the reaction time of participants to a lexical decision task and the distance of input-target pairs within the semantic network.

[3] Three other datasets were used: two related to concept-feature pairs, and a labeled word-association dataset cited as "Chen et al. (2024)", but whose reference is erroneous and cannot be found online. The dataset is reported to have a surprisingly high Cohen's κ (.81, twice as much as for WAX), but it cannot be found either.

Round	Sample	Meaning	Form	# pairs
1	1	.23	-	100
2	2	.39	-	100
3	3	.36	.45	50
	4	.65	.46	50
	5	.63	.58	50
4	6	.54	.55	50
	7	.54	.64	50
	8	.37	.60	50
5	9	.67	.55	50
	10	.75	.63	50
	11	.56	.62	50
3-5	3-11	.60	.58	450

Table 1: Inter-annotator agreement (Cohen's kappa) across annotation rounds for double-annotated samples.

model struggles with either the task, the taxonomy of word-association relations, or both.

In this paper, we propose a linguistically motivated inventory of cue-response relations meeting the requirements outlined in Section 2.1. We evaluate the relation taxonomy through inter-annotator agreement on a sample of English word associations extracted from SWOW. Additionally, we investigate word association classification using LLMs, both on our dataset and relation inventory, and on the WAX dataset and inventory (Liu et al., 2022). The relatively low performance observed for both leads us to discuss whether models have sufficient knowledge of the principles underlying human word associations.

3 A taxonomy of cue-response relations

This section presents the taxonomy we used to classify word-association relations, along with a labeled dataset of cue-response pairs. Crucially, we employed a dual-level classification, where the relation between a cue and a response is annotated for both meaning and form. We also took the directionality of the relations into account to reflect the asymmetry of the associations.

3.1 Methodology

We adopted an inductive approach to develop our taxonomic model, starting with basic linguistic categories that distinguish lexical relations, semantic features, argumental relations, and modification for the semantic part of the classification, and phonological and morphological relations for the formal part. These classes were explicitly defined and subsequently refined through multiple rounds of annotation and adjudication. Annotation guidelines were established, including a decision tree to systematize the annotation process[4].

Three expert annotators conducted double-blind annotation and adjudication over 5 rounds, on randomly selected samples from SWOW, focusing on associations between cues and R1 responses provided by at least three participants[5]. The annotation guidelines were revised after each round, and the process continued until a satisfactory inter-annotator agreement was reached. Sample sizes and inter-annotator agreement (IAA, Cohen's kappa scores) for each round of annotation are provided in Table 1.

The multi-level annotation with co-existing labels for meaning and form was introduced after Round 2, following the observation that a single label was insufficient to capture the complexity of association relations. For example, in the pair *pickup-truck*, the cue is a hyponym of the response (semantic label), and cue and response also form the compound word *pickup truck* (formal label). As can be seen in Table 1, the IAA increased significantly following the introduction of the two separate taxonomies (in Round 3), but remained stable in subsequent rounds, despite continued refinement of the annotation guidelines. Calculating IAA across all samples after taxonomy split (Rounds 3-5, Samples 3-11, totalling 450 instances), we obtained a Cohen's kappa of .60 for the Meaning taxonomy (34 labels) and .58 for Form (6 labels), representing a notable improvement over the results reported by Liu et al. (2022) for WAX ($\kappa = .42$, across 16 labels).

On top of the 450 double-annotated pairs, we sampled additional pairs from the SWOW dataset, annotated by a single expert. We obtained a dataset of 1,300 cue-response pairs, annotated for both form and meaning relations (hereafter the **F**orm **A**nd **M**eaning **W**ord **A**ssociations **(FAMWA)** dataset). The distribution of Meaning labels in FAMWA is provided in Figure 1[6]. Importantly, these 1,300 cue-response pairs were randomly sampled from the SWOW dataset, as an attempt to pre-

[4]The dataset and the guidelines are available at https://github.com/mariro8/FAMWA.

[5]We deliberately excluded words given only as second (R2) or third responses (R3) to better align with the standard single-word association task.

[6]The distribution for Form labels is quite skewed, with 1,070, 70, 98, 23, 23, and 16 items for 'none', 'compo_R+C', 'compo_C+R', 'in_mwe', 'similar', and 'morpho' labels, respectively.

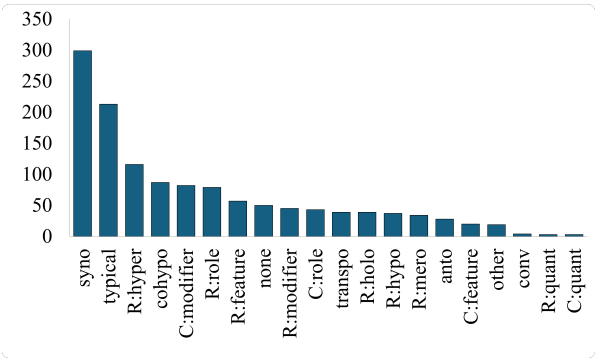


Figure 1: Distribution of Meaning labels in FAMWA. The various C/R:role_x categories are grouped into 2 single C/R:role categories, resulting in 20 labels.

serve the natural distribution of word association categories. This contrasts with the WAX dataset, as will be detailed when analyzing LLM classification in Section 4.

3.2 Resulting taxonomy

The final taxonomies for Form and Meaning consist of 34 and 6 categories, respectively (see Appendix B for the two lists of categories and their description). The Meaning taxonomy includes both lexical relations (i.e., synonymy, antonymy, hyponymy, etc.) and non-lexical relations (i.e., semantic features, semantic roles in predicate-argument relations[7], modifiers, etc.). The Form taxonomy distinguishes between phonological similarity, morphological relations (affixation or compounding), and multi-word expressions (when the cue and the response are part of a complex lexicalized expression involving other components). Both taxonomies include a "none" relation, which applies when the cue and response are unrelated in meaning or form, as well as an "other" label in the Meaning taxonomy to account for idiosyncratic relations.

Importantly, both the semantic and the formal categories are oriented, in order to capture asymmetrical relations. For instance, semantic roles were annotated when the cue is a typical argument of the response or vice-versa. A pair such as *promise-keep* was thus coded as C:role_theme, since the cue *promise* is the argument of *keep* with the role Theme, while the inverse pair would be analyzed as R:role_theme. Similarly, the sequences C+R and R+C were distinguished when cues and responses form compound words. For example, the pair *shopping-bag* was annotated as a C+R compound whereas the inverse pair would be classified as an R+C compound.

4 Ability of language models to classify associations

We now focus on examining the extent to which language-model-based systems are able to classify word associations, using human-designed linguistic classifications. Previous works have provided abundant evidence that LLMs have linguistic knowledge, and in particular lexical knowledge (see, e.g., Kello and Bruna 2024 and Hayashi 2025, who showed LLMs' ability to accurately detail lexical properties of words and to distinguish word senses in context). De Deyne et al. (2024) reported that GPT-4 performed poorly in classifying word associations with the WAX dataset and relation inventory, achieving an accuracy of only 47%. We hypothesize that a dataset with better IAA can lead to improved model performance. To test this hypothesis, we prompted various LLMs to label cue-response pairs and evaluated their performance on the FAMWA dataset. We also compared performance using the WAX dataset and taxonomy, which highlights the impact of category distribution.

4.1 Adjustments in taxonomies

To compare the classifications using taxonomies of roughly equal size, we merged some of the labels in our Meaning taxonomy—more precisely we merged all the C:role_x and R:role_x labels into C:role and R:role, respectively—and we dropped the labels with less than 10 instances[8], as well as the corresponding instances (10 instances in total). This resulted in a dataset used for classification of 1,290 instances (hereafter FAMWA-1290) with a Meaning relation inventory of 17 labels (shown as the first 17 bars in Figure 1), comparable in size to the 16 labels of the WAX taxonomy.

4.2 Models

Among the ever-growing list of available LLMs, we selected GPT-4o-mini, Llama.-3.1-70B and GPT-4o, namely three models of small, medium and large size.

[7]We used the semantic role tagset from Verbnet (Kipper et al., 2006), following the associated guidelines (https://verbs.colorado.edu/verb-index/VerbNet_Guidelines.pdf), and annotated semantic roles with the lowest possible role in the hierarchy.

[8]C/R:quant (C (resp. R) can be used as a quantifier of R (resp. C)) and conv (C and R are converse words)

4.3 Evaluation datasets

We tested the selected LLMs on FAMWA-1290 for Meaning labels and then proceeded to evaluate the best model only (GPT-4o) on other taxonomy/dataset pairs: FAMWA-1290 for Form labels, the complete labeled WAX (consisting of 1,602 instances), and the human-labeled WAX (725 instances). Notably, for more than half of the labeled instances, gold WAX labels were automatically obtained using patterns found in the explanations that participants provided during the word association task[9]. Since the automatically labeled instances are biased towards categories for which reliable patterns could be designed, they do not reflect the true distribution of categories among SWOW cue-responses. Hence, we also provide results on the human-labeled part of WAX.

4.4 Experimental protocol

We tested each model in three settings: zero-shot, few-shot (with exactly one example per label), as well as "implicit" few-shot, in which the description of a given category is accompanied by an example in parentheses. The same examples were used in few-shot and implicit few-shot settings, and throughout the experiments.

Our prompts included three distinct elements: the task description, the list of labels and their descriptions, and the input/output format. A fourth section is added in the few-shot setting, providing one example of input and expected output per label, in the desired format. We performed preliminary tests with a few formulations, for the task description and the input/output format (see the variants of each section in Appendix C). In these preliminary experiments, we tested 3 variants for the input/output format section (see Table 8 in Appendix C), which resulted in marginal performance differences. We then retained 4 formulations for the task description section, and a single formulation for the other sections of the prompt, resulting in 4 prompt variants which we tested in a systematic way for all the models, datasets and settings.

We used a zero temperature for all the experiments, hence forcing the models to always generate the most probable token at each position. When parsing the models' answers, we removed any symbols and converted the text to lowercase to match the answer with the label names. We counted the answers containing no known labels as incorrect.

In the few-shot and implicit few-shot settings, we removed the instances used as examples from the testing instances.

Labels: FAMWA Meaning			
Instances: FAMWA-1290			
Models	Zero	Impl.	Few
GPT-4o-mini	32.1 (4)	27.2 (2)	36.6 (1)
Llama-3.1-70B	41.0 (3)	41.2 (3)	42.3 (3)
GPT-4o	45.1 (3)	**46.2 (2)**	46.0 (1)
Labels: WAX			
Instances: full WAX (1602)			
GPT-4o	52.1 (3)	53.8 (4)	**57.5 (1)**
Instances: WAX human-labeled (725)			
GPT-4o	41.7 (1)	45.1 (3)	**46.1** (3)
Labels: FAMWA Form			
Instances: FAMWA-1290			
GPT-4o	**78.3 (1)**	76.6 (3)	75.0 (3)

Table 2: Accuracies for cue-response pair classification across datasets, models and settings (**zero**-shot, **implicit** few-shot, and **few**-shot). The best accuracy for the 4 prompt variants is reported, with their preferred task variant in the prompt indicated in parentheses (see Table 6 in Appendix C).

4.5 Results

The results of the experiment are provided in Table 2. Analysis of the results on FAMWA-1290 for Meaning labels reveals a consistent and expected trend: performance improves systematically with increasing **model size** across all prompt settings (zero, implicit few-shot, and few-shot). However, even in the best-performing setup—the implicit few-shot configuration—the accuracy reaches only 46.2%, showing that the overall performance of the best model is still limited.

Concerning **prompt settings**, providing examples, either in implicit few-shot or few-shot, systematically elicited better results than zero-shot. This suggests that, given the technical nature of the labels, the models struggle to "understand" them and their descriptions, and benefit from the inclusion of examples. In general, the implicit few-shot setting provided slightly lower performance than the few-shot approach, suggesting that the models take

[9]For instance, searching the pattern "opposite" within the explanations allowed Liu et al. (2022) to automatically classify 76 instances into the Antonym category.

advantage of examples presented in the expected input/output format. Still, this pattern did not apply to the FAMWA-1290 dataset with GPT-4o, since the best settings for Meaning and Form were the implicit few-shot and zero-shot, respectively.

We also compared the automatic classification of cue-response pairs between WAX and FAMWA-1290 Meaning inventories[10]. However, the comparison is limited by differences in evaluation instances and category distribution in the datasets. The classification task proved to be easier with the full set of WAX instances than with the FAMWA-1290 instances, since GPT-4o in few-shot setting achieved an accuracy of 57.5% on full WAX (vs. 46.2% on FAMWA-1290 Meaning labels in implicit few-shot). Yet, the performance dropped to 46.1% on the human-labeled part of WAX, which more accurately represents the distribution of categories found in SWOW cue-response pairs, similar to the performance on the FAMWA-1290 Meaning instances[11].

The accuracy was higher for the predictions of the FAMWA-1290 Form labels (78.3%), but this is largely attributable to the highly imbalanced distribution of classes, as will be discussed in the analysis of the performance across categories.

Performance across categories The overall accuracies presented in the previous section conceal substantial variation in both model performance and category prevalence. In this section, we investigate these differences on FAMWA-1290, as a dataset that reflects the category distribution observed in SWOW. Focusing on GPT-4o in the implicit few-shot setting, as the best-performing model and configuration, we examine the F1-score and number of instances for each of the FAMWA-1290 Meaning labels in Figure 2. The results show substantial variation across categories, ranging from F1 = 77.4 for the Antonym label to null performance for several others. It is worth noting that this performance ranking does not correspond to the frequency ranking in FAMWA (see Figure 1). For instance, Antonym was predicted more accurately than Synonym despite being ten times less frequent in FAMWA, while R:role, which appears

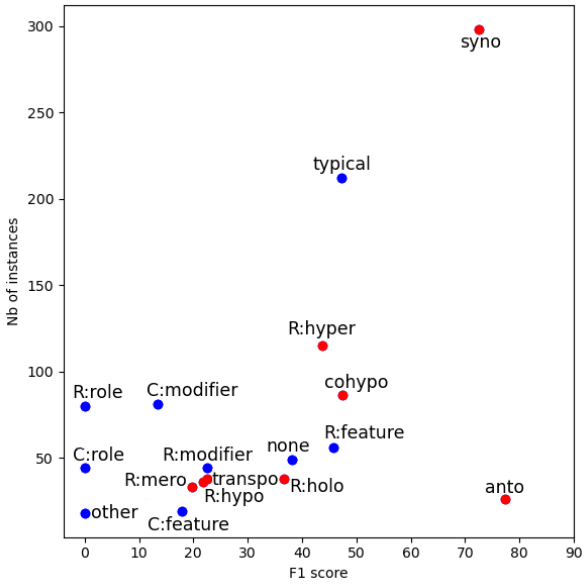


Figure 2: GPT-4o performance in the implicit setting on the FAMWA-1290 Meaning labels (excluding 17 instances used in prompt examples and 3 unpredicted relations). Lexical relations are shown in red and non-lexical relations in blue.

three times more often than Antonym (80 vs. 26 instances), was not predicted at all by GPT-4o.

The best classified relations were lexical ones (Antonym, Synonym, as well as R:hyper and Cohyponym to a lesser extent). Hyponymy was easier to detect when the response was a hypernym of the cue (F1 = 43.7 for R:hyper) than vice-versa (F1 = 21.8 for R:hypo), which is surprising given the reciprocal nature of the relation. This underscores the usefulness of using oriented categories when analyzing model performance. The Typical and R:feature categories are the only non-lexical relations that were predicted with moderate success (F1 $>$ 40%), whereas the performance on all other relations remains poor (F1 $<$ 40%). Moreover, in symmetric relations such as R:feature and C:feature, the R:x categories were consistently predicted more accurately than the C:x categories—for example, the prediction was better for R:feature and R:modifier than for C:feature and C:modifier. This suggests increased difficulty when the response is more central than the cue[12].

Turning to the breakdown by Form labels (Table 3), the model mostly predicted the absence of

[10]Note the WAX inventory does include a category "common phrase" which pertains to a formal classification, but most labels in WAX are semantic, hence the WAX inventory is more comparable to FAMWA Meaning than to FAMWA Form.

[11]De Deyne et al. (2024) obtained 47.1% on a fraction of the human-labeled WAX dataset when prompting an underspecified version of GPT-4.

[12]The breakdown per WAX label is provided in Table 9 in Appendix D, together with rough mappings with the FAMWA labels when applicable. It too shows varying performance across categories, and the same two best predicted categories (Antonym and Synonym).

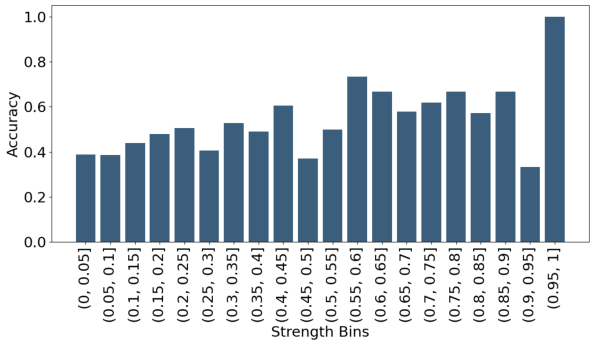


Figure 3: Accuracy of GPT-4o predicted Meaning labels, on FAMWA-1290, broken down across bins of associative strength of the pairs.

a formal relation—unsurprisingly given its prevalence in the dataset—but performed poorly across all other categories. We note though a relative ability to detect morphological relations, counterbalanced by a tendency to overpredict this category (P = 23.1, R = 1.0, F1 = 37.5). Moreover, identifying compounds proved more challenging for R+C compounds (F1 = 25.5) than for C+R compounds (F1 = 45.8), highlighting the model's difficulty in learning a pattern where the linguistic order (R then C) differs from that presented in the prompt (C then R).

Label	P	R	F1	Nb
none	90.3	88.0	89.1	1062
compo_C+R	63.6	35.7	45.8	98
morpho	23.1	1.0	37.5	15
similar	33.3	30.4	31.8	23
compo_R+C	34.1	20.3	25.5	69
in_mwe	5.6	17.4	8.4	23

Table 3: Performance of GPT-4o in zero-shot setting, for each of the FAMWA-1290 Form labels.

Performance across associative strengths We additionally examined whether cue-response pairs that are frequently provided by humans are easier for the models to classify. We used the concept of **associative strength** (De Deyne et al., 2019), defined for a cue-response pair (c, r) as the number of participants who gave r (as R1) in response to c, normalized for the total number of participants who provided at least one response for the cue c.

Figure 3 shows the performance of GPT-4o for the FAMWA-1290 Meaning labels, broken down by associative strength. There is no clear correlation between associative strength and classification accuracy, which varies across bins. However, bins with an associative strength above .55 generally exhibit higher accuracy compared to those with lower strength. Interestingly, this shift aligns with lexical relations surpassing non-lexical relations within the gold data[13]. Yet, it should be noted that the number of instances per bin, from where this shift happens up to the strongest bin, is lower than 20 (with the last 5 bins having between 3 and 10 instances). Consequently, any conclusions based on these bins should be interpreted cautiously due to the limited sample size.

5 Conclusion

Our efforts to develop a linguistically motivated taxonomy of word-association relations proved effective, as we achieved higher inter-annotator agreement than comparable studies using different analytical frameworks. Integrating a dual-level analysis of formal and semantic relations, while also accounting for directionality in associations, is not only more satisfactory from the perspective of linguistic description, but also ensures greater stability and consistency in annotation quality, at least with expert annotators. Nevertheless, the observed agreement remains moderate, highlighting the inherent challenge of producing a metalinguistic analysis of word associations. This difficulty is frequently noted by researchers who attempt to classify word associations, and it contrasts with the naturalness of the word association task itself, which is effortless as it relies only on the existence of the mental lexicon. Arguably, the basic task of generating word associations and the metalinguistic task of analyzing them involve fundamentally distinct cognitive processes and engage contrasting aspects of the language faculty.

While expert human annotators achieve only moderate agreement, even advanced models like GPT-4o exhibit mediocre performance in analyzing semantic relations between cues and responses, despite their well-documented linguistic capacities. This is congruent with the previous conclusion that the word association task and its analysis leverage different types of capabilities. Moreover, LLMs' ability to classify word associations is not improved by refinements in descriptive frameworks and varies considerably across relation classes. The

[13]The bin with strength between .90 and .95 is the only exception, both in accuracy and number of non-lexical relations.

limited performance of LLMs in labeling word associations should be analyzed in light of their ability to produce them. The inherent heterogeneity and variability of word associations pose challenges not only for their metalinguistic analysis, but also for their generation by language models. Future research should explore this generative capacity in greater depth, for example through detailed analysis of the LLM-generated association norms reported in LWOW (Abramski et al., 2025).

Acknowledgments

We are grateful to three anonymous reviewers for their careful reading and constructive comments that helped us refine the arguments presented in this paper.

References

Maria A. Rodriguez and Paola Merlo. 2020. Word associations and the distance properties of context-aware word embeddings. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 376–385, Online. Association for Computational Linguistics.

Katherine Abramski, Riccardo Improta, Giulio Rossetti, and Massimo Stella. 2025. The "LLM World of Words" English free association norms generated by large language models. *Scientific Data*, 12(1):803.

Marianna Bolognesi, Roosmaryn Pilgram, and Romy Van Den Heerik. 2017. Reliability in content analysis: The case of semantic feature norms classification. *Behavior Research Methods*, 49:1984–2001.

Simon De Deyne, Chunhua Liu, and Lea Frermann. 2024. Can GPT-4 recover latent semantic relational information from word associations? A detailed analysis of agreement with human-annotated semantic ontologies. In *Proceedings of the Workshop on Cognitive Aspects of the Lexicon @ LREC-COLING 2024*, pages 68–78, Torino, Italia. ELRA and ICCL.

Simon De Deyne, Danielle J Navarro, Amy Perfors, Marc Brysbaert, and Gert Storms. 2019. The "Small World of Words" English word association norms for over 12,000 cue words. *Behavior research methods*, 51:987–1006.

James Deese. 1962. Form class and the determinants of association. *Journal of verbal learning and verbal behavior*, 1(2):79–84.

James Deese. 1965. *The structures of associations in language and thought*. The John Hopkins Press.

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.

Tess Fitzpatrick. 2006. Habits and rabbits: Word associations and the l2 lexicon. *EUROSLA yearbook*, 6(1):121–145.

Tess Fitzpatrick. 2007. Word association patterns: unpacking the assumptions. *International Journal of Applied Linguistics*, 17:319–331.

Francis Galton. 1879. Psychometric experiments. *Brain*, 2(2):149–162.

Murray Glanzer. 1962. Grammatical category: A rote learning and word association analysis. *Journal of verbal learning and verbal behavior*, 1(1):31–41.

Yoshihiko Hayashi. 2025. Evaluating LLMs' capability to identify lexical semantic equivalence: Probing with the word-in-context task. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6985–6998, Abu Dhabi, UAE. Association for Computational Linguistics.

Carl G Jung. 1910. The association method. *The American journal of psychology*, 21(2):219–269.

Cristopher Kello and Polyphony J. Bruna. 2024. Emergent mental lexicon functions in ChatGPT. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46, pages 5452–5459.

Grace Helen Kent and Aaron Joshua Rosanoff. 1910. A study of association in insanity. *American Journal of Psychiatry*, 67(1):37–96.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2006. Extending VerbNet with novel verb classes. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

George R Kiss, Christine Armstrong, Robert Milroy, and James Piper. 1973. An associative thesaurus of English and its computer analysis. *The computer and literary studies*, 153.

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.

Chunhua Liu, Trevor Cohn, Simon De Deyne, and Lea Frermann. 2022. WAX: A new dataset for word association explanations. 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 1: Long Papers)*, pages 106–120.

Ken McRae, Saman Khalkhali, and Mary Hare. 2012. Semantic and associative relations in adolescents and young adults: Examining a tenuous dichotomy.

Paul Meara. 1983. Word associations in a foreign language. *Nottingham Linguistics Circular*, 11(2):29–38.

Douglas L Nelson, Cathy L McEvoy, and Thomas A Schreiber. 2004. The University of South Florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, 36(3):402–407.

Klaus F Riegel and Irina WM Zivian. 1972. A study of inter-and intralingual associations in English and German 1. *Language Learning*, 22(1):51–63.

Ava Santos, Sergio E Chaigneau, W Kyle Simmons, and Lawrence W Barsalou. 2011. Property generation reflects word association and situated simulation. *Language and Cognition*, 3(1):83–119.

Amos Tversky. 1977. Features of similarity. *Psychological review*, 84(4):327.

Špela Vintar, Mojca Brglez, and Aleš Žagar. 2024. How human-like are word associations in generative models? An experiment in Slovene. In *Proceedings of the Workshop on Cognitive Aspects of the Lexicon@ LREC-COLING 2024*, pages 42–48.

Leticia Vivas, M Yerro, Sofía Romanelli, A García Coni, Ana Comesaña, F Lizarralde, I Passoni, and J Vivas. 2022. New Spanish semantic feature production norms for older adults. *Behavior Research Methods*, pages 1–17.

Sabine Schulte im Walde, Alissa Melinger, Michael Roth, and Andrea Weber. 2008. An empirical characterisation of response types in german association norms. *Research on Language and Computation*, 6(2):205–238.

R. S. Woodworth. 1938. Experimental psychology: Association. *American Psychological Association*, pages 340–367.

Ling-ling Wu and Lawrence W Barsalou. 2009. Perceptual simulation in conceptual combination: Evidence from property generation. *Acta psychologica*, 132(2):173–189.

Limitations

While this study provides insights into the classification of word associations, some limitations should be acknowledged.

First, we used exclusively English, which restricts the generalizability of the findings to other languages. Word associations are known to be influenced by linguistic and cultural aspects, thus, it can be interesting to explore whether the proposed taxonomy and observed patterns hold across different languages.

Second, the dataset used was relatively small, consisting of only 1,300 instances. While this allowed for detailed annotation and analysis, expanding the dataset would improve the robustness of the taxonomy and enable more reliable evaluation of model performance.

Finally, this study was conducted on a small scale due to the limited availability of expert annotators and the highly time-consuming nature of the annotation process. The reliance on a small group of experts, while ensuring high-quality annotations, may introduce biases or limit the diversity of perspectives in the classification process.

A Examples of form-meaning annotation for word associations

cue	response	meaning	form
banana	yellow	R:feature	none
capita	per	none	compo_R+C
stone	wall	C:modifier	compo_C+R
rareness	rarity	syno	morpho
meet	greet	typical	in_mwe
weight	height	cohypo	similar
lottery	win	C:role_theme	none

Figure 4: Examples of Meaning and Form annotations for word associations.

B Form taxonomy and Meaning taxonomy

We detail the FAMWA inventory for Form (Table 4) and Meaning (Table 5) labels with their descriptions. Short labels were used in the human annotation while their extended form was used for prompts.

C Prompts for labeling word associations

We provide below the various forms of prompts we tested on Llama-3.1-8B/70B. A single prompt is made of a description of task (4 variants shown in Table 6), the list of labels with their description (Table 7), and the description of the expected output format (Table 8).

D GPT-4o performance on WAX human-labeled dataset broken down per WAX label

We detail the performance of GPT-4o on the human-labeled part of the WAX dataset (725 instances). We provide the WAX label and the corresponding FAMWA label, if a mapping is possible, with the F1-score performance and the total number of instances per class (see Table 9).

Short labels	Prompt labels	Definitions	Examples
compo_C+R	compound cue response	the sequence C R forms a compound	hermit-crab
compo_R+C	compound response cue	the sequence R C forms a compound	rights-human
in_mwe	multiword expression	R and C belong to the same multiword expression, containing other elements	pedal-metal
morpho	morphological relation	R and C belong to the same derivational or inflectional paradigm	gave-gift
similar	similar	C and R are similar in graphical or phonological form but not morphologically related	hat-cat
none	no relation	No formal relationship	roof-house

Table 4: FAMWA Form inventory: short labels, corresponding labels used in prompts, short definitions, and corresponding examples.

Short labels	Prompt labels	Definitions	Examples
syno	synonym	The cue and the response are synonyms	belly-stomach
anto	antonym	The cue and response are antonyms	large-small
transpo	transposition	The cue and the response are synonyms but they have a different part-of-speech	smelly-stink
cohypo	cohyponym	The cue and the response have a close common hypernym but they are not synonyms	weight-height
typical	typical	There is an obvious (and implicit) predicate that links the cue and the response or when the eventuality denoted by the cue typically cooccurs with that denoted by the response (or vice versa)	honey-bee
R:hyper	hypernym	The response is a hypernym of the cue	labrador-dog
R:hypo	hyponym	The response is a hyponym of the cue	pets-cat
R:holo	holonym	The response is a holonym of the cue	roof-house
R:mero	meronym	The response is a meronym of the cue	universe-stars
C:feature	response's feature	The cue is a semantic feature of the response	green-grass
R:feature	cue's feature	The response is a semantic feature of the cue	sauna-hot
C:modifier	response's modifier	The cue is used as a modifier of the response	metallic-paint
R:modifier	cue's modifier	The response is used as a modifier of the cue	debt-school
C:role_x	response's argument	C is an argument of R with semantic role x	pillow-sleep
R:role_x	cue's argument	R is an argument of C with semantic role x	hike-woods
C:quant	response's quantifier	C can be used as a quantifier of R	bunch-grapes
R:quant	cue's quantifier	R can be used as a quantifier of C	item-one
conv	converse	C and R are converse words	prey-predator
other	other relation	The cue and the response are in a semantic relation of different type than those listed above	trip-vacation
none	no relation	No semantic relation between cue and response	shall-we

Table 5: FAMWA Meaning inventory: short labels, corresponding labels used in prompts, short definitions and corresponding examples. C/R:quant and conv labels were discarded in the evaluation.

Tasks	Description
Variant 1	Objective: Given a word association, consisting of a pair of cue and response, label the semantic relation between these pairs with a label based on the specified criteria.
Variant 2	Objective: Given a word association task where a cue word elicits a response word, classify the semantic relation between the cue word and the response word using one of the labels described in the specified criteria.
Variant 3	Objective: You're a linguist interested in semantic relations between words. Given a pair of words, a cue word and a response word, classify the semantic relation between the cue and the response using one of the labels described in the specified criteria.
Variant 4	Objective: You're a linguist interested in semantic relations between words. Given a pair of words, composed by a cue and a response, classify the pair into its corresponding semantic relation using the labels described in the specified criteria.

Table 6: Variants for the first section of the prompts: description of the task to perform.

Criteria:
Synonym: The cue and the response are synonyms (CUE:recycle, RESPONSE:reuse)
Antonym: The cue and the response are antonyms (CUE:outside RESPONSE:inside)
Hypernym: The response is a hypernym of the cue (CUE:piano, RESPONSE:instrument)
Hyponym: The response is a hyponym of the cue (CUE:mammal, RESPONSE:human)
Meronym: The response is a meronym of the cue (CUE:face, RESPONSE:nose)
Holonym: The response is a holonym of the cue (CUE:plant, RESPONSE:garden)
Transposition: The cue and the response are synonyms but they have a different part-of-speech (CUE:anger, RESPONSE:mad)
Cohyponym: The cue and the response have a common hypernym but they are not synonyms (CUE:discourse, RESPONSE:conversation)
Response's argument: The cue is a syntactic argument of the response (CUE:rabbit, RESPONSE:hop)
Cue's argument: The response is a syntactic argument of the cue (CUE:filled, RESPONSE:cup)
Response's feature: The cue is a semantic feature of the response (CUE:explosive, RESPONSE:dynamite)
Cue's feature: The response is a semantic feature of the cue (CUE:sunset, RESPONSE:orange)
Response's modifier: The cue is used as a modifier of the response (CUE:custard, RESPONSE:pudding)
Cue's modifier: The response is used as a modifier of the cue (CUE:friend, RESPONSE:best)
Typical: There is an obvious (and implicit) predicate that links the cue and the response or when the eventuality denoted by the cue typically cooccurs with that denoted by the response (or vice versa) (CUE:incense, RESPONSE:church)
No relation: No semantic relation between the cue and the response (CUE:rally, RESPONSE:pep)
Other relation: The cue and the response are in a semantic relation of different type than those listed in our labels (CUE:saucy, RESPONSE:sauce)

Table 7: Section 2 of the prompts, listing the labels, each accompanied by a description, and an example. The examples are provided only in the "implicit few-shot" setting.

Type format	Description
Format 1	Input and Output format: The input follows the format: 'Input: CUE:cue_word, RESPONSE:response_word' where cue_word is the cue word and response_word is the response word. The output follows the format: 'Output: CUE:cue_word, RESPONSE:response_word, LABEL:label' where cue_word is the cue word and response_word is the response word and label is one of the labels in the specified criteria. Generate only the content without explanations following strictly the output format.
Format 2	Input and Output format: The input follows the format: 'CUE:cue_word, RESPONSE:response_word' where cue_word is the cue word and response_word is the response word. The output follows the format: 'CUE:cue_word, RESPONSE:response_word, LABEL:label' where cue_word is the cue word and response_word is the response word and label is one of the labels in the specified criteria. Generate only the content without explanations following strictly the output format.
Format 3	Input and Output format: The input is a pair of words that follows the format: 'CUE:cue_word - RESPONSE:response_word' where cue_word is the cue word and response_word is the response word. As output, return the corresponding label. Generate only the content without explanations following strictly the output format.

Table 8: Variants for the Section 3 of the prompts: desired input/output formats. The Format 3 variant was retained after tests on the Llama models.

Label (Corresp.)	**F1**	**Nb**
antonym (= Antonym)	55	8
synonym (= Synonym)	78	122
material made of ($\subset$ R/C:modifier)	40	2
has property ($\subset$ R/C:feature)	69	81
location ($\subset$ R/C:role)	34	43
category exemplar ($\subset$ R:hyper+R:hypo)	35	42
function	44	52
part of ($\subset$ (R:holo+R:mero))	37	38
common phrase	52	69
action ($\subset$ R:role)	27	104
emotion evaluation	26	18
time	46	19
result in	25	49
has prerequisite	23	22
thematic ($\subset$ R:role)	23	44
same category (= cohyponym)	18	12

Table 9: GPT-4o F1-scores and number of instances in the human-labeled WAX dataset (for each WAX label). Rough correspondence to FAMWA labels is indicated in parentheses, when applicable.

Computational Semantics Tools for Glue Semantics

Mark-Matthias Zymla
University of Konstanz
mark-matthias.zymla
@uni-konstanz.de

Mary Dalrymple
University of Oxford
mary.dalrymple
@ling-phil.ox.ac.uk

Agnieszka Patejuk
Institute of Computer Science
Polish Academy of Sciences
aep@ipipan.waw.pl

Abstract

This paper introduces a suite of computational semantic tools for Glue Semantics, an approach to compositionality developed in the context of Lexical Functional Grammar (LFG), but applicable to a variety of syntactic representations, including Universal Dependencies (UD). The three tools are: 1) a Glue Semantics prover, 2) an interface between this prover and a platform for implementing LFG grammars, and 3) a system to rewrite and add semantic annotations to LFG and UD syntactic analyses, with a native support for the prover. The main use of these tools is computational verification of theoretical linguistic analyses, but they have also been used for teaching formal semantic concepts.

1 Introduction

This paper introduces a suite of tools related to Glue Semantics (Dalrymple 1999, Asudeh 2022, 2023), an approach to compositionality based on the idea of resource sensitivity, for a wider computational semantic audience.[1] On this approach, the compositional process is not necessarily determined directly by phrasal constituency (as in, for example, Heim and Kratzer 1998), but is rather guided by pairing (partial) semantic representations with linear logic formulas referring to parts of syntactic representations. While Glue Semantics has been most extensively applied in the context of Lexical Functional Grammar (LFG; Kaplan and Bresnan 1982, Bresnan et al. 2015, Dalrymple et al. 2019, Dalrymple 2023), it has also been successfully combined with other syntactic formalisms, including Universal Dependencies (e.g., Gotham and Haug 2018), Lexicalized Tree Adjoining Grammar (Frank and van Genabith 2001), Head-driven Phrase Structure Grammar (Asudeh and Crouch 2002), and Minimalism (Gotham 2018). It is compatible with various formal meaning representations, including predicate logic with lambdas and DRT (Kamp and Reyle 1993).

Within LFG, computational research evolves around the Xerox Linguistics environment (XLE; Crouch et al. 2017), a platform that has been primarily tailored towards the modeling of syntax. Although XLE grammars are being developed all across the world, the investigation of semantic issues in LFG from a computational perspective received impetus with the introduction of an early version of the Glue Semantics Workbench (GSWB; Meßmer and Zymla 2018).[2] This paper presents new contributions to GSWB and two recently developed resources that make use of it.[3]

The central resource presented in this paper is the Glue Semantics Workbench (GSWB), a modular system for calculating Glue Semantics (henceforth, Glue) proofs. It provides three different Glue provers and is designed to permit the implementation of additional provers based on varying linear logic fragments and meaning languages (e.g., predicate logic with lambdas, DRT, etc.).

The second tool, XLE+Glue, implements an interface between GSWB and XLE.[4] This tool allows users to specify semantic contributions of lexical items and syntactic rules in XLE grammars, which can then be fed into GSWB for semantic calculation. The system has been mainly developed to explore what is called a "co-descriptive approach" to Glue (explained in §2.2). XLE+Glue also illustrates the possibility of GSWB to work with different meaning languages.

[1] Early versions of two of these tools have been presented LFG-internally, the third is presented here for the first time.

[2] Earlier works in computational semantics related to LFG include Asher and Wada 1988, Crouch 2005, Crouch and King 2006, Bobrow et al. 2007, Lev 2007.

[3] See §3 for links to Github repositories of these resources.

[4] The original idea is presented in Dalrymple et al. 2020. This paper presents further developments.

Proceedings of the 16th International Conference on Computational Semantics, pages 199–217
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

The third tool presented in this paper is a system for linguistic graph expansion and rewriting (LiGER). It is inspired by the original XLE transfer system, which was initially used for machine translation (Frank 1999) and later mainly for semantic parsing (Crouch 2005, Crouch and King 2006), but also as a full-fledged reasoning engine (Bobrow et al. 2007), indicating its versatility. LiGER has been developed because the original transfer component of XLE is no longer supported by XLE. Like the original transfer system, LiGER can be used to enrich XLE analyses with information from other linguistic resources. With respect to semantic analysis, it provides the possibility of exploring the second major approach to deriving Glue representations, "description-by-analysis" (see §2.2), and thus complements XLE+Glue.

Overall, the tools presented here allow researchers to experiment with different settings within the Glue framework, including the choice of a suitable linear logic fragment, the choice of meaning language, and the choice of co-description vs. description-by-analysis approaches to deriving meaning representations. The goal of this paper is to illustrate the capabilities of these tools and how they can be used for verifying theoretical analyses and for exploring formal semantic concepts. Section 2 explains the LFG architecture, focusing on two aspects: the projection structure and Glue. Section 3 describes the three tools in more detail, while §4 mentions some use cases. Section 5 concludes.

2 Background

Within the LFG community, the development of XLE grammars, as well as associated resources such as treebanks, is carried out mainly in the scope of the Parallel Grammar (ParGram) project (Butt et al. 2002, Sulger et al. 2013). Such grammars have been developed for a wide variety of typologically diverse languages, demonstrating the cross-linguistic and formal validity of LFG's (morpho)syntactic component.[5] The work presented in this paper aims to facilitate extending such syntactic work to semantics. This section first describes the underlying concepts of the LFG formalism, and then the LFG approach to semantics.

[5] Some of the grammars that are publicly available for testing via INESS (https://clarino.uib.no/iness/xle-web; Rosén et al. 2012), and some that are not yet publicly available (in parentheses), are:

(i) Larger grammars for English, German, French, Norwegian, and Polish (as well as Chinese and Japanese)

(ii) Smaller grammars for Georgian, Indonesian, Malagasy, Turkish, Welsh, Wolof, and Urdu (as well as Greek and Hungarian)

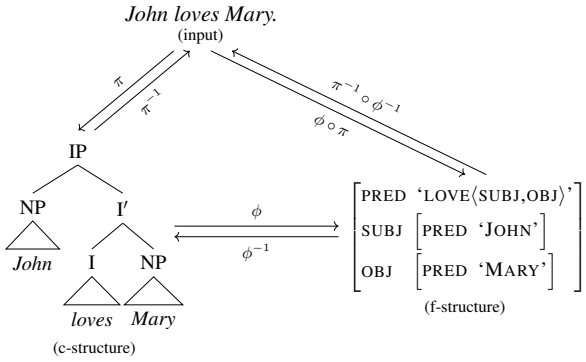


Figure 1: LFG correspondence structure as implemented in XLE

2.1 LFG projection architecture

LFG is developed around the idea of mutually constraining parallel representations. The two syntactic representations, implemented in XLE, are c(onstituent)-structure and f(unctional)-structure (cf. Figure 1). While c-structure encodes the surface structure in terms of a constituent parse that preserves linear word order, f-structure encodes functional information, primarily grammatical functions and morphosyntactic features, in an attribute-value matrix. Grammars encode both structures simultaneously. C-structures are constrained by phrase structure rules (as in the first row in (1)), with categories specified in lexical entries (see "N" in (2)). F-structures are constrained using functional annotations (usually equations) in phrase-structure rules and lexical entries.

(1) IP → NP I′
(↑ SUBJ) = ↓ ↑ = ↓

(2) John N (↑ PRED) = 'JOHN'

This simultaneous specification of two levels is called local co-description (Bresnan et al. 2015). In this architecture, the different structures are related via projection functions. This ensures structural correspondence between different levels of analysis and entails mutual accessibility of projections.

Consider Figure 1. The c-structure is generated from the input via the π-projection – a constituent parse. The f-structure is specified based on constraints that are annotated on c-structure nodes and specified in the lexicon. The corresponding mapping function from c- to f-structure is encoded in the ϕ-projection. The mapping from the input to

f-structure is a combination of the two projections: the $\phi \circ \pi$ mapping.[6] LFG also assumes an inverse of each mapping function; while such inverse mappings are less often discussed, they play a role in the possibility of generation, as explored in early work within XLE.[7]

The next section discusses two ways of integrating semantics into the LFG projection architecture.

2.2 Semantics in LFG

Adding any projection that preserves the kind of bi-directionality described in the previous section to this framework is a challenge, and this also holds for the semantic projection. It is beyond the scope of this paper to delve into all the fine details of semantics in LFG, but we briefly address some of the main challenges that the tools presented here may help address. For this purpose, we first provide a quick introduction to Glue, which is a semantic formalism that has been developed for LFG but is generally applicable to different linguistic frameworks, making the present tools interesting for projects that go beyond LFG as well.

The formalism of Glue is modeled around the idea of resource sensitivity (Dalrymple 1999). Resource management is ensured by the use of a fragment of the resource-sensitive *linear logic* (Girard 1987) that is paired with a meaning representation, forming a *meaning constructor*. Example (3) shows meaning constructors for all words in *John loves Mary*. In this example, each word in the sentence introduces a single meaning constructor.[8] In (3), the meaning representation j of the subject *John* is associated with the resource g, the meaning m of the object *Mary* is associated with the resource h, and the more complex meaning of the verb *loves* is associated with the linear logic formula $g \multimap (h \multimap f)$. This formula uses the linear implication $\multimap$ to indicate that it requires the resource in its antecedent to produce the resource in its consequent. Thus, by consuming the subject resource g, we can produce the resource $(h \multimap f)$, which in turn consumes the object resource h to produce the final result f corresponding to the meaning of the full sentence; see the full Glue proof in (4). In line with the Curry-Howard isomorphism (CHI), modus ponens on the linear logic side corresponds to function application on the meaning side.

(3)
$$\begin{array}{ll} \textit{John} & j : g \\ \textit{Mary} & m : h \\ \textit{loves} & \lambda x.\lambda y.love(x,y) : g \multimap (h \multimap f) \end{array}$$

(4)
$$\dfrac{\dfrac{\lambda x.\lambda y.love(x,y) : g \multimap (h \multimap f) \quad j : g}{\lambda y.love(j,y) : h \multimap f} \quad m : h}{love(j,m) : f}$$

This relation between resource consumption and semantic composition is the foundation of Glue. As long as the CHI is preserved, different fragments of linear logic can be paired with different meaning representations, resulting in two dimensions of variation.

Additionally, as mentioned above, Glue has been combined with different syntactic theories, assuming different approaches to the syntax/semantics interface. In this paper, we briefly discuss the two main such approaches explored in LFG: co-description (Kaplan and Wedekind 1993) and description by analysis (Halvorsen and Kaplan 1988).

In the co-descriptive approach, particular to LFG, meaning constructors are introduced in lexical entries (and, possibly, grammatical rules), in parallel with categorical and functional information. This is illustrated on the left-hand side of Figure 2. The lexical entries use the ↑-variable to refer to specific elements in the f-structure. The nominal entries specify the semantics for the substructures they contribute (corresponding to g and h at the bottom of the figure). The inflected verb uses the functional descriptions $(\uparrow \textsc{subj})$ and $(\uparrow \textsc{obj})$ to retrieve these substructures via their indices to form the meaning constructor of the verb.

On the other hand, description-by-analysis uses a fully assembled f-structure as input to derive meaning constructors. This is usually done by rules that match partial f-structure descriptions and introduce corresponding meaning constructors; see the right-hand side of Figure 2. There, `#f`, `#g`, and `#h` are variables referring to f-structures (see the corresponding f, g, and h at the bottom of Figure 2),

[6] The projection structure is usually depicted in linear order on a form-to-meaning mapping (Kaplan 1995, Asudeh 2006); however, to avoid directionality, we present the projection structure as a (complete) graph, with no order between nodes since the order might well change depending on specific processing tasks (Jackendoff 2010).

[7] Both parsing and generation are in principle undecidable in LFG and require additional constraints on the formalism to be made workable (Kaplan and Bresnan 1982). See Wedekind 1988 for early LFG work on generation from a separate semantic structure, i.e., involving an inverse of the mapping from semantics to the surface string, and Wedekind and Kaplan 2020 and Kaplan and Wedekind 2019 for more recent work. Such work motivates the existence of inverse projection mappings, and such mappings are assumed in this paper.

[8] This is not a rule; a word can introduce any number of meaning constructors, and meaning constructors may also be introduced by syntactic rules.

co-description:

John	N	$(\uparrow \text{PRED}) = \text{`JOHN'}$ $j : \uparrow$
Mary	N	$(\uparrow \text{PRED}) = \text{`MARY'}$ $m : \uparrow$
loves	I	$(\uparrow \text{PRED}) = \text{`LOVE}\langle\text{SUBJ,OBJ}\rangle\text{'}$ $\lambda x.\lambda y.love(x, y) :$ $(\uparrow \text{SUBJ}) \multimap ((\uparrow \text{OBJ}) \multimap \uparrow)$

description-by-analysis:

```
#f SUBJ #g PRED %g  ==>  #g GLUE %g : #g.

#f OBJ #h PRED %h  ==>  #h GLUE %h : #h.

#f SUBJ #g & #f OBJ #h & #f PRED %f
    ==>  #f GLUE %f : #g -o (#h -o #f).
```

result (for both approaches):

$$f\begin{bmatrix}\text{PRED} & \text{`LOVE}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & g\big[\text{PRED `JOHN'}\big] \\ \text{OBJ} & h\big[\text{PRED `MARY'}\big]\end{bmatrix} \qquad \begin{array}{l} j : g \\ m : h \\ \lambda x.\lambda y.love(x, y) : g \multimap (h \multimap f)\end{array}$$

Figure 2: Co-descriptive lexicon vs. description-by-analysis rules

used as resources in the linear logic side of the introduced meaning constructors, while `%f`, `%g`, and `%h` refer to the corresponding PRED values and are used in the meaning sides. The first two rules introduce resources for the subject and the object, while the rule for the verb specifies the meaning constructor in a way similar to the co-descriptive approach. This means that both approaches generally map the same kind of nodes onto meaning constructors as indicated by the f-structure and the corresponding instantiated meaning constructors to its right (see the indices g, h, and f there).

Both co-description and description-by-analysis are currently in use in theoretical LFG work; it might well be the case that it is best to combine the two approaches to deal with different kinds of semantic phenomena.[9] The present tool suite is designed to allow for this.

2.3 Semantic autonomy

The flexibility in modeling the syntax/semantics interface is due to one of the key advantages of Glue Semantics: a high level of semantic autonomy (Asudeh 2004). As Figure 2 suggests, semantic composition does not rely on word order – it relies instead on more general concepts such as grammatical functions. Furthermore, semantic autonomy provides a purely semantic treatment of quantification, one that is independent of syntactic considerations such as, for instance, quantifier raising (Heim and Kratzer 1998). This is illustrated in Figure 3 on the basis of quantifier scope ambiguity. For a more in-depth discussion on quantifier scope, see, e.g., Gotham (2019, 2021), Dalrymple et al. (1999). Semantic autonomy provides a unique view on formal semantics that can be explored using the tools presented in this paper.

[9] It seems that description-by-analysis may be more suitable for the semantic interpretation of functional features, whereas phenomena involving information structure are more suitably encoded in a co-descriptive fashion (Andrews 2008).

2.4 Related work

The tools presented here are inspired by work in grammar engineering (e.g., Flickinger et al. 2017) and semantic annotation (e.g., Basile et al. 2012). There is also some overlap with toolkits such as the NLTK (Bird et al. 2009). The main difference is a focus on Glue Semantics and its compositional properties, as well as its relation to various syntactic approaches, especially LFG and Universal Dependencies (UD). The present tools have not yet been employed in large-scale grammar engineering efforts, but rather at the interface between formal and computational linguistics to verify analyses (but see Zymla et al. 2025, Findlay et al. 2023).

3 The tools

The ParGram project provided a cross-linguistically informed approach to syntactic and semantic parsing, though the latter was mostly worked out for English, while concrete implementations for other languages were of limited scope. This is largely due to the fact that the semantics relied heavily on various external resources that were not available cross-linguistically. Semantic parsing relied on ordered rewriting rules implemented as part of a transfer system in XLE (Crouch and King 2006, Bobrow et al. 2007). Another important issue addressed with the present tools is that the existing transfer system is neither publicly available nor compatible with the currently available XLE releases provided by

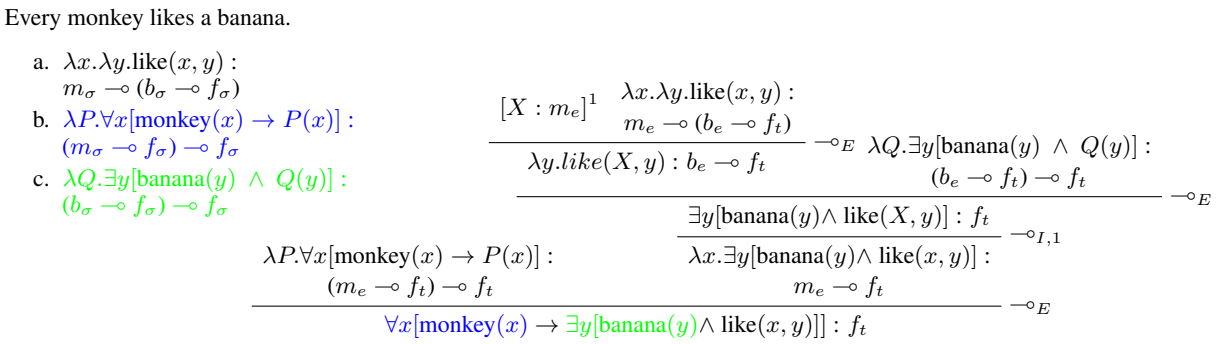


Figure 3: **Quantification in Glue:** Quantifier scope falls out naturally from the properties of linear logic, giving appropriate typings. Implication introduction (lambda abstraction) allows to capture flexible scope configurations (the alternative reading for this example is shown in Figure 7 in appendix A).

the University of Konstanz.[10] The tools described below are open source and compatible with various systems, including XLE, and they are designed to be useful in theoretical linguistic work as well as in investigation of general issues of integrating semantics into the LFG projection architecture.

3.1 The Glue Semantics Workbench

The Glue Semantics Workbench (GSWB)[11] is a modular system for deriving Glue proofs. To this end, it provides the possibility of using different provers as well as different input formats for meaning languages, with a built-in parser for formulas based on typed lambda-calculus, and support for meaning representations written in Prolog (in particular, those developed on the basis of Blackburn and Bos 2005, i.e., untyped lambda calculus and λ-DRT). Furthermore, functionality was recently added that allows users to interface GSWB with NLTK's (Bird et al. 2009) semantic capabilities (Klein 2006).

GSWB uses a string format for linear logic and semantic representations that is close to actual Glue semantic representations, as illustrated in (5).

(5)
```
john : g
mary : h
[/x_e.[/y_e.love(x,y)]] :
    (g -o (h -o f))
```

There, the meaning side is on the left of :, and the linear logic side is on the right. The entry for the verb shows the encoding of complex linear logic formulas and lambda expressions which can be computed using the basic tools for function application (Blackburn and Bos 2005).

To ensure flexibility, the meaning side of a meaning constructor can be replaced with any semantic representation that can be encoded as a string. In this case, users can specify procedures that preserve CHI, by implementing function application directly in GSWB or by feeding the output to a separate system.[12] The latter option is used to integrate GSWB with a modified version of the DRT part of Boxer tools (Bos 2008; based on Blackburn and Bos 2005) and with NLTK (Findlay et al. 2023).

GSWB contains three different provers for the implicational fragment of linear logic: one with linear quantification (prover 1) and two variants of a prover without linear quantification (prover 2). Both variants of prover 2 are based on Hepple 1996 and Lev 2007, but one is extended with a notation for conducting multistage proving (Findlay and Haug 2022), a process that essentially allows for the grouping of meaning constructors to constrain the order of application. This is one way of accounting for restrictions on scope-taking expressions like quantifiers, embedding verbs, etc.

These provers provide separate additional functionalities for exploring the resulting Glue derivations, including reasons why a derivation might fail. Specifically, prover 1 has two functionalities. First, it allows for a depth-first search of intermediate results in a failed proof, extracting those partial solutions that would need to be combined to find a successful proof. Second, it allows the proofs to be given in natural deduction form. This is illustrated in Figure 4 based on (5).

The two variants of prover 2 also allow users to visualize a derivation. More specifically, they

[10] ling.sprachwiss.uni-konstanz.de/pages/xle/

[11] https://github.com/Mmaz1988/GlueSemWorkbench_v2

[12] For a string *a* corresponding to a function and an argument string *b*, the default procedure produces the string *a(b)*.

```
[/x e.[/y e.love(x,y)]] : (g -o (h -o f))     john : g
-------------------------------------------------------E
        [/x e.[/y e.love(x,y)]](john) : (h -o f)          mary : h
        ---------------------------------------------------------E
                [/x e.[/y e.love(x,y)]](john)(mary) : f
```

Figure 4: Natural deduction proof by GSWB, based on meaning constructors in (5)

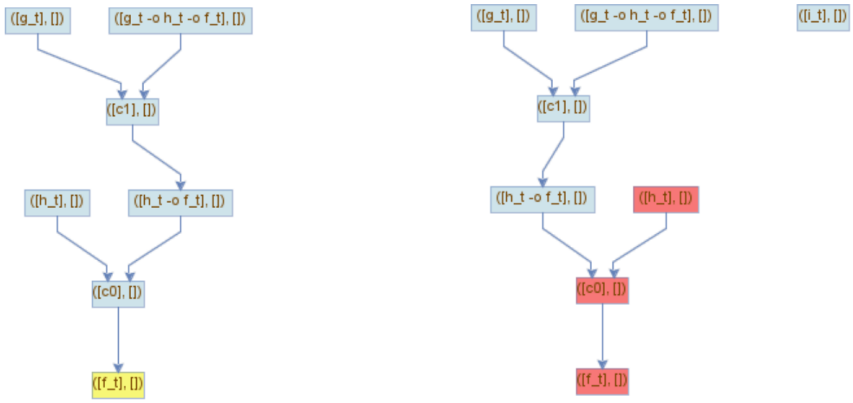


Figure 5: **Successful derivation graph for the proof in (4) and an alternative failed derivation graph**: The graph on the left presents input meaning constructors and combination steps as blue nodes and highlights the goal category in yellow. The graph on the right is based on an erroneous input that is superficially similar to (4) . Missing resources (leaves of the graph) and failed derivation steps are marked in red so as to make it easier to debug the proof. The proof fails since h is required by the verb as a resource corresponding to the object. However, in this unsuccessful proof the object was assigned the resource i, which is a dangling node since it has no consumer.

produce a derivation graph. This graph roughly corresponds to a proof tree but highlights cyclic elements in the derivation (indicating compositional ambiguities), if present (cf. Lev 2007: ch. 6). Figure 5 illustrates the visualization. (The derivation there does not have any cyclic elements.)[13]

Current and future developments of GSWB are mainly geared toward the interpretability of the output of GSWB, as illustrated in Figures 4–5, as well as the integration in broader processing pipelines. This is illustrated by reference to the next two tools, which use the capabilities presented above.

3.2 XLE+Glue

XLE+Glue has been developed as an interface between XLE and GSWB corresponding to LFG+Glue in the theoretical literature. It is integrated into the XLE user interface and can be used out of the box.

The original version[14] consists of a specification for Glue meaning constructors in terms of attribute-value matrices that can be represented as part of f-structures (Dalrymple et al. 2020).

Example (7) illustrates the encoding of the Glue meaning constructor in (6) as an AVM in an f-structure. As shown there, linear logic resources are added via the GLUE attribute, whose value is a set of semantic representations. These are described in terms of AVMs encoding their MEANING side (simply a string corresponding to the meaning) and their linear logic side. The latter uses nested expressions to reflect linear implication: ARG1 and ARG2 refer to linear logic resources (not semantic arguments) that need to be consumed to produce the resource f with type t.

(6) $\quad love : g_e \multimap (h_e \multimap f_t)$

(7)
$$f \left[\begin{array}{ll} \text{PRED} & \text{`LOVE<SUBJ,OBJ>'} \\ \text{SUBJ} & g\,[\,] \\ \text{OBJ} & h\,[\,] \\ \text{GLUE} & \left\{\begin{array}{l} \left[\begin{array}{ll} \text{MEANING} & \text{LOVE} \\ \text{ARG1} & \left[\begin{array}{ll} \text{RESOURCE} & g \\ \text{TYPE} & e \end{array}\right] \\ \text{ARG2} & \left[\begin{array}{ll} \text{RESOURCE} & h \\ \text{TYPE} & e \end{array}\right] \end{array}\right] \\ \left[\begin{array}{ll} \text{RESOURCE} & f \\ \text{TYPE} & t \end{array}\right] \end{array}\right\} \end{array}\right]$$

[13] While this example is trivial, finding errors in more complex proofs can be difficult, especially when manually working with the GSWB.

[14] https://github.com/Mmaz1988/xle-glueworkbench-interface

More recently, a version with an alternative notation for meaning constructors has been developed[15] that is closer to their representation in formal semantic theory. The alternative notation is similar to that of GSWB but uses references to f-structure nodes, as in Figure 2 on the left. This is illustrated in (8).

(8)
```
[/x_e.[/y_e.P(x,y)]]:
((^SUBJ)_e -o ((^OBJ)_e -o ^_t))
```

While the notation is different, the implementation boils down to the idea of the original XLE+Glue. However, now, when loading a grammar in XLE, meaning constructors written as in (8) are automatically translated into AVM representations by a script, making the grammars leaner. Furthermore, such meaning constructors may be easier to read than the nested templates necessary to encode meaning constructors in the original approach.

This approach is, in principle, an implementation of the co-descriptive approach to Glue since the templates are generally called from the lexicon. The XLE+Glue repository provides several sample XLE grammars containing templates that produce the corresponding meaning constructors. These grammars exhibit the various parameters along which XLE+Glue can be tweaked: it allows for exploring different meaning languages (currently, first-order logic and λ-DRT), and it enables the user to specify meaning constructors in the f-structure or in a separate semantic structure. Furthermore, although the current paper presents XLE+Glue as a venue for exploring co-descriptive approaches to Glue, it is, in fact, more flexible, since the Glue AVMs corresponding to meaning constructors need not be specified in the lexicon. They could be specified via rewrite rules or, possibly, in other ways. However, since it is the only resource in this paper making a concrete proposal for exploring semantic co-description, it is unique in this regard.

On the technical side, XLE+Glue consists of an extension to the XLE user interface and a translation component that rewrites the specified meaning constructors into a format compatible with GSWB.[16] Thus, XLE+Glue is, essentially, an interface between XLE and GSWB.

[15] https://github.com/Mmaz1988/xleplusglue

[16] The original translation component was written in Prolog. For the new system, the scripts have been moved to a Java implementation.

3.3 Linguistic Graph Expansion and Rewriting

The Linguistic Graph Expansion and Rewriting (LiGER)[17] tool allows for the specification of rules that rewrite and expand f-structure nodes, as shown in Figure 2 on the right. The system is based on graph matching techniques, but also provides tools to check for certain LFG-specific relations such as (inside-out) functional uncertainty. The graphs are described in terms of queries inspired by corpus search engines, in particular the one designed for LFG within INESS (Rosén et al. 2012; https://clarino.uib.no/iness/). Before querying, the system translates f-structures into more general graph structures. This mechanism is inspired by the original XLE transfer system (Crouch et al. 2017, Ide and Bunt 2010), but it is applicable beyond the annotations provided by the XLE. For example, it provides an interface to the Stanford Universal Dependency parser (Manning et al. 2014). Generally speaking, it is mainly geared towards the analysis of directed (acyclic) graphs that underlie many syntactic analyses.

Figure 6 illustrates normalization from syntactic representations to directed graphs. Given this kind of normalization, the system can be combined with various linguistic resources to either specify structural correspondences or expand graphs with additional information. The primary use of the system is currently the specification of semantic rules inspired by the description-by-analysis tradition in Glue (Kaplan and Wedekind 1993). It combines insights from computational approaches, e.g., Crouch 2005 and Crouch and King 2006, with more recent theoretical approaches (Andrews 2008, 2010). The former employ a destructive approach during which a given f-structure is taken as input to a set of ordered rewrite rules. These rules incrementally consume parts of the f-structure to produce semantic constraints, sometimes involving intermediate representations and access to external resources (e.g., for lexical semantics). Thus, the inverse mapping from semantics to syntax is not trivially recoverable.[18] By contrast, the theoretical approach involves working towards a structure-preserving implementation, i.e., a monotonic approach to description-by-analysis, more clearly maintaining LFG's bi-directionality. This choice is

[17] https://github.com/Mmaz1988/abstract-syntax-annotator-web

[18] See Zarrieß and Kuhn (2010) for discussion.

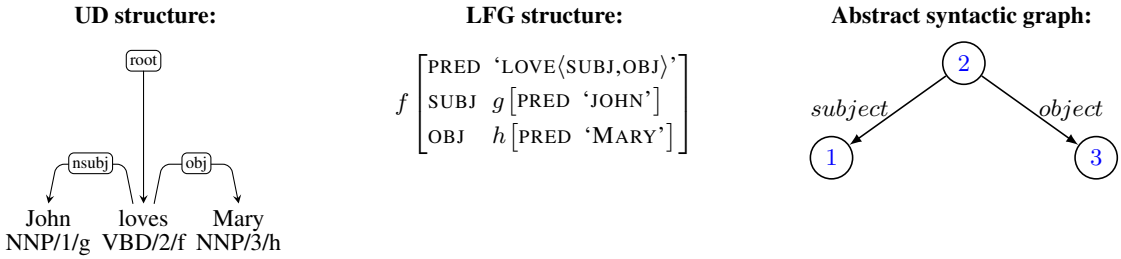


Figure 6: Parallelized syntax for: *John loves Mary*

not constrained by LiGER, but rather by how the system is used. Thus, it is well-suited to explore the notion of description-by-analysis.

LiGER is implemented in Java as an application and a web service in parallel, so it can be used in web-based applications and more traditional annotation pipelines. As indicated above, it is compatible with Universal Dependencies (as provided by Stanford CoreNLP) and XLE representations. It can also be used to call the corresponding parsers from their respective resources.

4 Use cases

At this stage of development, XLE+Glue and LiGER have not been widely used for broad coverage semantic parsing (but see Findlay et al. 2023 for a broad coverage use of the GSWB). However, they have already been employed for verification of theoretical LFG+Glue analyses (see §4.1), for a teaching grammar (see §4.2), and for research on ambiguity management (see §4.3).

4.1 Verification of theoretical analyses

The tools described above have been used to verify theoretical analyses. For example, GSWB has been employed in an investigation of scope interactions between nominal and verbal quantifiers (Zymla and Sigwarth 2019), LiGER in an analysis of Greek tense and aspect (Zymla and Fiotaki 2021), and XLE+Glue in an account of gapping (Przepiórkowski and Patejuk 2023).

In particular, Przepiórkowski and Patejuk 2023 propose a theoretical LFG+Glue analysis of gapping, as in English *Marge saw Lisa and Homer Bart*, with the second conjunct meaning 'Homer saw Bart'. The analysis crucially relies on Champollion's (2015) compositional treatment of event semantics and is relatively complex, to the extent that it is not trivial to manually verify its predictions for more complex cases, such as (9), which is expected to have the two readings in (10)–(11).

(9) Tracy introduced Lisa to Marge and Bart to Homer.

(10) $[\exists e.\, introduce(e) \wedge agent(e,t) \wedge theme(e,l) \wedge beneficiary(e,m)] \wedge [\exists e.\, introduce(e) \wedge \underline{agent(e,t)} \wedge \underline{theme(e,b)} \wedge beneficiary(e,h)]$

'Tracy introduced Lisa to Marge and Tracy introduced Bart to Homer.'

(11) $[\exists e.\, introduce(e) \wedge agent(e,t) \wedge theme(e,l) \wedge beneficiary(e,m)] \wedge [\exists e.\, introduce(e) \wedge \underline{agent(e,b)} \wedge \underline{theme(e,l)} \wedge beneficiary(e,h)]$

'Tracy introduced Lisa to Marge and Bart introduced Lisa to Homer.'

However, using XLE+Glue, the formal analysis was implemented as an XLE grammar and all reading were derived automatically. In the case of (9), they all turned out to be equivalent to (10) or (11).

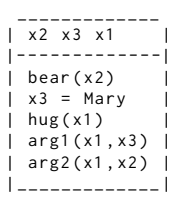
4.2 Teaching grammar

A different application of the presented suite of Glue tools concerns a teaching grammar implementing analyses of some phenomena encountered in a grammar development class, especially tense and aspect.

Using GSWB and LiGER, the grammar produces DRT representations based on the Boxer tools exemplifying a Neo-Davidsonian event semantics. An example is shown in (12). There, *x1* refers to an event with two arguments, *x2* and *x3*. These are enumerated based on an argument hierarchy (Bresnan and Kanerva 1989). For the purpose of this paper, *arg1* generally refers to an agentive role, *arg2* refers to a theme/patient role, and *arg3* generally refers to a recipient/goal role.[19]

(12) Mary hugged a bear.

[19]Thus, the argument roles are comparable to those in the PropBank (Palmer et al. 2005), but they are not verb-specific.



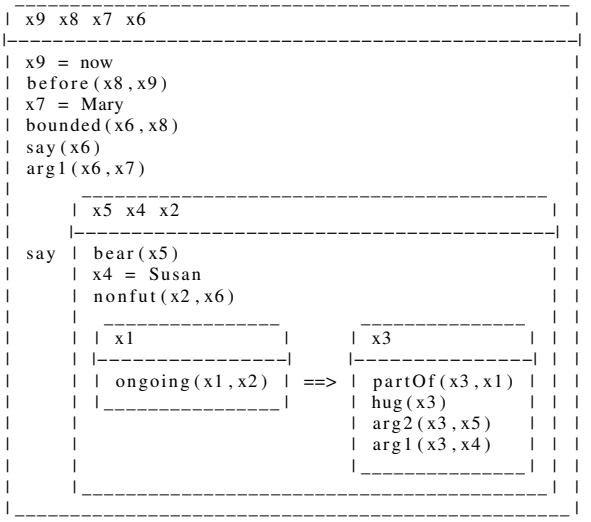
Appendix B contains additional examples of DRSs produced by the grammar on the basis of meaning constructors derived by LiGER from f-structures.

For example, we have added to the grammar some basic LiGER rules for tense/aspect interpretation. In this case, the importance of LiGER lies in contextualizing tense/aspect features according to their morphosyntactic context and beyond. This is illustrated in example (13) below, where the interpretation of the embedded tense is constrained by the matrix tense. To put it concisely, the embedded tense must be evaluated relative to the matrix tense and it may only be evaluated as simultaneous or anterior to it as indicated by the two readings of (13) shown in (14)–(15). This is explained in more detail in Zymla 2017, 2018. Furthermore, this example illustrates differences between perfective (bounded) and imperfective (ongoing) grammatical aspect in the matrix clause and embedded clause respectively (see Zymla 2019 for details).

(13) Mary said that Susan was hugging a bear.

(14) Mary said: Susan is hugging a bear.

(15) Mary said: Susan was hugging a bear.



As these examples show, the teaching grammar combined with LiGER allows us to capture important syntactic and semantic generalizations and, thus, explore substantial insights into the interplay between syntax and semantics. Importantly, the grammar also illustrates the distinction between co-description and description-by-analysis pointed out in the previous sections.

4.3 Exploring the cross-linguistic variability of semantic ambiguities

Moot and Retoré (2012) point out that semantic composition does not fall into one neatly categorized logic but rather moves on a spectrum of constrainedness of the underlying logic. Works like Gotham 2019, 2021, building on Barker 2022, explore subtle cross-linguistic differences in the flexibility of scope-taking semantic operators that highlight this issue. Constraints on semantic composition have been explored in the context of computational grammars and Glue semantics in Findlay and Haug 2022 and Zymla 2024 using the GSWB. This research not only improves the formal adequacy of LFG, but potentially also allows for more precise statements about the logic of composition more generally and its interaction with other modules of grammar, particularly, from a cross-linguistic perspective. Relatedly, Butt et al. (2024) present an approach to disambiguating questions via prosodic information that is computationally implemented in combination with XLE+Glue.

5 Summary

This paper has presented a suite of computational tools for Glue Semantics. It has established their relevance for exploring various important concepts, particularly how to embed a semantic component in the LFG projection architecture as implemented within XLE. In this regard, both a co-descriptive and a description-by-analysis approach have been presented, covering the two major proposals for semantic analysis in LFG.

The tools presented in this paper contribute to a growing ecosystem of LFG-based CL tools which are actively developed on multiple fronts. They receive regular updates and new features.

However, it is not only the LFG community that may benefit from these tools. Glue Semantics is compatible with various kinds of syntactic and semantic analyses. From the perspective of NLP, one of the most interesting prospects may be its compatibility with Universal Dependencies (Haug and Findlay 2023), but even beyond that, the tools presented in this paper provide an exciting avenue for research on formal semantics in computational linguistics.

References

Avery D Andrews. 2008. The role of PRED in LFG + Glue. In *Proceedings of the LFG08 Conference*, pages 47–67.

Avery D Andrews. 2010. Propositional glue and the correspondence architecture of LFG. *Linguistics and Philosophy*, 33(3):141–170.

Nicholas Asher and Hajime Wada. 1988. A computational account of syntactic, semantic and discourse principles for anaphora resolution. *Journal of Semantics*, 6(1):309–344.

Ash Asudeh. 2004. *Resumption as resource management*. Ph.D. thesis, Stanford University.

Ash Asudeh. 2006. Direct compositionality and the architecture of LFG. *Intelligent linguistic architectures: Variations on themes by Ronald M. Kaplan*, pages 363–387.

Ash Asudeh. 2022. Glue semantics. *Annual Review of Linguistics*, 8:321–341.

Ash Asudeh. 2023. Glue semantics. In (Dalrymple 2023), pages 651–697.

Ash Asudeh and Richard Crouch. 2002. Glue semantics for HPSG. In *Proceedings of the 8th international HPSG conference, Stanford, CA. CSLI Publications*.

Chris Barker. 2022. Rethinking scope islands. *Linguistic Inquiry*, 53(4):633–661.

V. Basile, J. Bos, K. Evang, and N. Venhuizen. 2012. A platform for collaborative semantic annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 92–96, Avignon, France.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing — Analyzing text with Python and the Natural Language Toolkit*. O'Reilly.

Patrick Blackburn and Johannes Bos. 2005. *Representation and inference for natural language: A first course in computational semantics*. Center for the Study of Language and Information Amsterdam.

Daniel G. Bobrow, Bob Cheslow, Cleo Condoravdi, Lauri Karttunen, Tracy Holloway King, Rowan Nairn, Valeria de Paiva, Charlotte Price, and Annie Zaenen. 2007. PARC's bridge and question answering system. In *Proceedings of the GEAF 2007 Workshop*, pages 1–22.

Johan Bos. 2008. Wide-coverage semantic analysis with Boxer. In *Semantics in text processing. step 2008 conference proceedings*, pages 277–286.

Joan Bresnan, Ash Asudeh, Ida Toivonen, and Stephen Wechsler. 2015. *Lexical-functional syntax*, volume 16. John Wiley & Sons.

Joan Bresnan and Jonni M Kanerva. 1989. Locative inversion in Chichewa: A case study of factorization in Grammar. *Linguistic inquiry*, pages 1–50.

Miriam Butt, Tina Bögel, Mark-Matthias Zymla, and Benazir Mumtaz. 2024. Alternative questions in Urdu: From the speech signal to semantics. In *Proceedings of the LFG'24 Conference*, Konstanz. PubliKon.

Miriam Butt, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. The parallel grammar project. In *Proceedings of the 2002 Workshop on Grammar Engineering and Evaluation*, volume 15, pages 1–7. Association for Computational Linguistics.

Lucas Champollion. 2015. The interaction of compositional semantics and event semantics. *Linguistics and Philosophy*, 38(1):31–66.

Dick Crouch, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, John T. Maxwell III, and Paula Newman. 2017. *XLE Documentation*. Palo Alto Research Center.

Richard Crouch. 2005. Packed rewriting for mapping semantics to KR. In *Proceedings of the Sixth International Workshop on Computational Semantics (IWCS-6)*, pages 103–114, Tilburg.

Richard Crouch and Tracy Holloway King. 2006. Semantics via f-Structure rewriting. In *Proceedings of the LFG06 Conference*, pages 145–165, Stanford, CA. CSLI Publications.

Mary Dalrymple. 1999. *Semantics and syntax in Lexical Functional Grammar: The resource logic approach*. MIT Press.

Mary Dalrymple, editor. 2023. *Handbook of Lexical Functional Grammar*. Language Science Press, Berlin.

Mary Dalrymple, John Lamping, Fernando Pereira, and Vijay Saraswat. 1999. Quantification, anaphora, and intensionality. In Mary Dalrymple, editor, *Semantics and Syntax in Lexical Functional Grammar – The Resource Logic Approach*, pages 39–89.

Mary Dalrymple, John J. Lowe, and Louise Mycock. 2019. *The Oxford reference guide to Lexical Functional Grammar*. Oxford University Press, Oxford.

Mary Dalrymple, Agnieszka Patejuk, and Mark-Matthias Zymla. 2020. XLE+Glue – A new tool for integrating semantic analysis in XLE. In *Proceedings of the LFG'20 Conference, Australian National University*, Stanford, CA. CSLI Publications.

Jamie Findlay and Dag Haug. 2022. Managing scope ambiguities in Glue via multistage proving. In *Proceedings of the Lexical Functional Grammar Conference*, pages 144–163.

Jamie Y Findlay, Saeedeh Salimifar, Ahmet Yıldırım, and Dag TT Haug. 2023. Rule-based semantic interpretation for Universal Dependencies. In *Proceedings of the Sixth Workshop on Universal Dependencies (UDW, GURT/SyntaxFest 2023)*, pages 47–57.

Dan Flickinger, Stephan Oepen, and Emily Bender. 2017. Sustainable development and refinement of complex linguistic annotations at scale. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, pages 353–377. Springer.

Anette Frank. 1999. From parallel grammar development towards machine translation. A project overview. *Proceedings of Machine Translation Summit VII" MT in the Great Translation Era*, pages 134–142.

Anette Frank and Josef van Genabith. 2001. GlueTag: Linear logic based semantics for LTAG – and what it teaches us about LFG and LTAG. In *The Proceedings of the LFG'01 Conference*, pages 104–126, University of Hong Kong. CSLI Publications.

Jean-Yves Girard. 1987. Linear logic. *Theoretical Computer Science*, 50(1):1 – 101.

Matthew Gotham. 2018. Making logical form type-logical: Glue Semantics for minimalist syntax. *Linguistics and Philosophy 41(5)*, pages 411–556.

Matthew Gotham. 2019. Constraining scope ambiguity in LFG+Glue. In *Proceedings of the LFG'19 Conference*, pages 111–129, Stanford, CA. CSLI Publications.

Matthew Gotham. 2021. Approaches to scope islands in LFG+Glue. In *Proceedings of the LFG'21 Conference*, pages 146–166, Stanford, CA. CSLI Publications.

Matthew Gotham and Dag Trygve Truslew Haug. 2018. Glue semantics for Universal Dependencies. In *The Proceedings of the LFG'18 Conference*, pages 208–226, Stanford, CA. CSLI Publications.

Per-Kristian Halvorsen and Ronald M. Kaplan. 1988. Projections and semantic description in Lexical-Functional Grammar. In *Proceedings of the International Conference on Fifth Generation Computer Systems, FGCS 1988, Tokyo, Japan, November 28-December 2, 1988*, pages 1116–1122. OHMSHA Ltd. Tokyo and Springer-Verlag.

Dag TT Haug and Jamie Y Findlay. 2023. Formal semantics for dependency grammar. In *Proceedings of the Seventh International Conference on Dependency Linguistics (Depling, GURT/SyntaxFest 2023)*, pages 22–31.

Irene Heim and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Blackwell, Malden, MA.

Mark Hepple. 1996. A compilation-chart method for linear categorial deduction. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 537–542. Association for Computational Linguistics.

Nancy Ide and Harry Bunt. 2010. Anatomy of annotation schemes: Mapping to GrAF. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 247–255.

Ray Jackendoff. 2010. The parallel architecture and its place in cognitive science. In Bernd Heine and Heiko Narrog, editors, *The Oxford Handbook of Linguistic Analysis*, pages 583–605. Oxford: Oxford University Press.

Hans Kamp and Uwe Reyle. 1993. *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and Discourse Representation Theory*, volume 42. Springer Science & Business Media.

Ronald M Kaplan. 1995. Three seductions of computational psycholinguistics. *Formal Issues in Lexical-Functional Grammar*, 47.

Ronald M. Kaplan and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281. MIT Press, Cambridge, MA.

Ronald M Kaplan and Jürgen Wedekind. 1993. Restriction and correspondence-based translation. In *Sixth Conference of the European Chapter of the Association for Computational Linguistics*.

Ronald M Kaplan and Jurgen Wedekind. 2019. Tractability and discontinuity. In *Proceedings of the International Lexical-Functional Grammar Conference*, pages 130–148.

Ewan Klein. 2006. Computational semantics in the Natural Language Toolkit. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 26–33.

Iddo Lev. 2007. *Packed computation of exact meaning representations*. Ph.D. thesis, Stanford University.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Moritz Meßmer and Mark-Matthias Zymla. 2018. The Glue Semantics Workbench: A Modular Toolkit for Exploring Linear Logic and Glue Semantics. In *Proceedings of the LFG'18 Conference, University of Vienna*, pages 249–263, Stanford, CA. CSLI Publications.

Richard Moot and Christian Retoré. 2012. *The logic of categorial grammars: A deductive account of natural language syntax and semantics*. Number 6850 in Lecture Notes in Computer Science. Springer, Heidelberg.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Adam Przepiórkowski and Agnieszka Patejuk. 2023. Filling gaps with Glue. In *The Proceedings of the LFG'23 Conference*, pages 223–240. PubliKon.

Victoria Rosén, Koenraad De Smedt, Paul Meurer, and Helge Dyvik. 2012. An open infrastructure for advanced treebanking. In *META-RESEARCH Workshop on Advanced Treebanking at LREC2012*, pages 22–29.

Sebastian Sulger, Miriam Butt, Tracy Holloway King, Paul Meurer, Tibor Laczkó, György Rákosi, Cheikh M Bamba Dione, Helge Dyvik, Victoria Rosén, Koenraad De Smedt, Agnieszka Patejuk, Özlem Çetinŏglu, I Wayan Arka, and Meladel Mistica. 2013. ParGramBank: The ParGram parallel treebank. In *ACL*, pages 550–560.

Jurgen Wedekind. 1988. Generation as structure driven derivation. In *Coling Budapest 1988 Volume 2: International Conference on Computational Linguistics*.

Jürgen Wedekind and Ronald M Kaplan. 2020. Tractable Lexical-Functional Grammar. *Computational Linguistics*, 46(3):515–569.

Sina Zarrieß and Jonas Kuhn. 2010. Reversing f-structure rewriting for heneration from meaning representations. In *Proceedings of the 15th International Conference on Lexical-Functional Grammar (LFG10)*, pages 479–499, Ottawa, Canada. CSLI Publications.

Mark-Matthias Zymla. 2017. Comprehensive annotation of cross-linguistic variation in the category of Tense. In *12th International Conference on Computational Semantics*.

Mark-Matthias Zymla. 2018. Annotation of the syntax/semantics interface as a bridge between deep linguistic parsing and TimeML. In *Proceedings 14th Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, pages 53–59.

Mark-Matthias Zymla. 2019. Aspectual reasoning in LFG – A computational approach to grammatical and lexical aspect. In *Proceedings of the LFG'19 Conference, Australian National University*, pages 353–373, Stanford, CA. CSLI Publications.

Mark-Matthias Zymla. 2024. Ambiguity management in computational Glue semantics. In *Proceedings of the LFG'24 Conference*, pages 285–310, Konstanz, Germany. PubliKon.

Mark-Matthias Zymla and Alexandra Fiotaki. 2021. Perfective non-past in Modern Greek. In *Proceedings of the LFG'21 Conference, On-Line*, pages 332–352, Stanford, CA. CSLI Publications.

Mark-Matthias Zymla, Kascha Kruschwitz, and Paul Zodl. 2025. Semantic parsing and reasoning in LFG – the case of gradable adjectives. In *Proceedings of the BriGap-2 Workshop: Bridges and Gaps between Formal and Computational Linguistics*. To appear.

Mark-Matthias Zymla and Gloria Sigwarth. 2019. On the syntax/semantics interface in computational Glue Semantics: A case study. In *Proceedings of the LFG'19 Conference, Australian National University*, pages 374–392, Stanford, CA. CSLI Publications.

A Additional proofs

$$\dfrac{\lambda P.\forall x[\text{monkey}(x) \to P(x)] : (m_e \multimap f_t) \multimap f_t \qquad \dfrac{\dfrac{\dfrac{[X : m_e]^1 \quad \lambda x.\lambda y.\text{like}(x,y) : m_e \multimap (b_e \multimap f_t)}{\lambda y.\text{like}(X,y) : b_e \multimap f_t} \multimap_E \quad [Y : b_e]^2}{\text{like}(X,Y) : f_t} \multimap_E}{\lambda x.\text{like}(x,Y) : m_e \multimap f_t} \multimap_{I,1}}{\dfrac{\forall x[\text{monkey}(x) \to \text{like}(x,Y)] : f_t}{\lambda y.\forall x[\text{monkey}(x) \to \text{like}(x,y)] : b_e \multimap f_t} \multimap_{I,2} \qquad \lambda Q.\exists y[\text{banana}(y) \wedge Q(y)] : (b_e \multimap f_t) \multimap f_t}{\exists y[\text{banana}(y) \wedge \forall x[\text{monkey}(x) \to \text{like}(x,y)]] : f_t} \multimap_E$$

Figure 7: Glue proof: *Every monkey likes a banana* inverse scope

B Worked out examples

(16) Mary hugged a bear.

```
 _____________
| x2  x3  x1  |
|-------------|
| bear(x2)    |
| x3 = Mary   |
| hug(x1)     |
| arg1(x1,x3) |
| arg2(x1,x2) |
|_____________|
```

Produced meaning constructors:

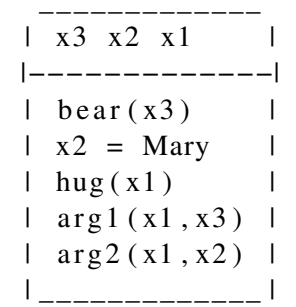
```
{
lam(V,lam(X,lam(E,merge(app(V,E),drs([],[rel(arg2,E,X)])))))  :
    ((6_v -o 6_t) -o (4_e -o (6_v -o 6_t))) || noscope
lam(X,drs([],[eq(X,'Mary')]))  :  (8_e -o 8_t)
lam(X,drs([],[pred(bear,X)]))  :  (4_e -o 4_t)
lam(V,lam(X,lam(E,merge(app(V,E),drs([],[rel(arg1,E,X)])))))  :
    ((6_v -o 6_t) -o (8_e -o (6_v -o 6_t))) || noscope
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X)))))  :
    ((8_e -o 8_t) -o ((8_e -o 5_t) -o 5_t)) || noscope
lam(V,drs([],[pred(hug,V)]))  :  (6_v -o 6_t)
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X)))))  :
    ((4_e -o 4_t) -o ((4_e -o 5_t) -o 5_t))
lam(V,merge(drs([E],[]),app(V,E)))  :  ((6_v -o 6_t) -o 5_t)
}
```

F-structure:

```
"Mary hugged a bear"

      [PRED     'hug<[1:Mary], [26:bear]>'                          ]
      [       1[PRED 'Mary'                                         ]]
      [SUBJ  66[CASE nom, GEND fem, NTYPE name, NUM sg, PERS 3      ]]
      [      68[                                                    ]]
      [                                                             ]
      [      35[PRED 'bear'                                         ]]
      [     102[                                                    ]]
      [OBJ   26[SPEC [DET [PRED 'a']]                               ]]
  14  [      34[CASE acc, DEF -, NTYPE count, NUM sg, PERS 3        ]]
  78  [     104[                                                    ]]
 108  [                                                             ]
 122  [TNS-ASP [MOOD indicative, PERF -_, PROG -_, TENSE past]      ]
 124  [PASSIVE -                                                    ]
```

(17) Mary was hugged by a bear.



Produced meaning constructors:

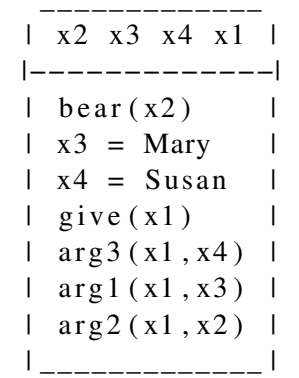
```
{
lam(V,merge(drs([E],[]),app(V,E))) : ((6_v -o 6_t) -o 5_t)
lam(V,lam(X,lam(E,merge(app(V,E),drs([],[rel(arg2,E,X)]))))) :
    ((6_v -o 6_t) -o (8_e -o (6_v -o 6_t))) || noscope
lam(X,drs([],[eq(X,'Mary')])) : (8_e -o 8_t)
lam(X,drs([],[pred(bear,X)])) : (4_e -o 4_t)
lam(V,lam(X,lam(E,merge(app(V,E),drs([],[rel(arg1,E,X)]))))) :
    ((6_v -o 6_t) -o (4_e -o (6_v -o 6_t))) || noscope
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X))))) :
    ((8_e -o 8_t) -o ((8_e -o 5_t) -o 5_t)) || noscope
lam(V,drs([],[pred(hug,V)])) : (6_v -o 6_t)
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X))))) :
    ((4_e -o 4_t) -o ((4_e -o 5_t) -o 5_t))
}
```

F-structure:

```
"Mary was hugged by a bear"

      [PRED      'hug<[38:bear], [1:Mary]>'                                  ]
               1 [PRED 'Mary'                                                 ]
      SUBJ    87 [CASE nom, GEND fem, NTYPE name, NUM sg, PERS 3              ]
              89 [                                                           ]
              56 [PRED 'bear'                                                ]
             132 [SPEC [DET [PRED 'a']]                                      ]
              47 [                                                           ]
      OBL-AG  55 [DEF -, NTYPE count, NUM sg, PERS 3, PFORM by, PTYPE nosem  ]
    26       134 [                                                           ]
   101        38 [                                                           ]
    14        46 [                                                           ]
    15       138 [                                                           ]
   140                                                                        ]
   150 TNS-ASP   [PERF -_, PROG -_, TENSE past]                               ]
   152 PARTICIPLE past, PASSIVE +                                             ]
```

(18) Susan was given the bear by Mary.



Produced meaning constructors:

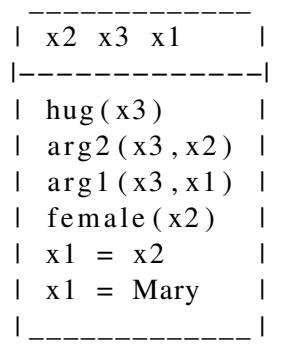
```
{
lam(V,merge(drs([E],[]),app(V,E))) : ((4_v -o 4_t) -o 3_t)
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X))))) :
    ((6_e -o 6_t) -o ((6_e -o 3_t) -o 3_t)) || noscope
lam(V,lam(X,lam(E,merge(app(V,E),drs([],[rel(arg3,E,X)]))))) :
    ((4_v -o 4_t) -o (8_e -o (4_v -o 4_t))) || noscope
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X))))) :
    ((2_e -o 2_t) -o ((2_e -o 3_t) -o 3_t)) || noscope
lam(X,drs([],[pred(bear,X)])) : (2_e -o 2_t)
lam(V,lam(X,lam(E,merge(app(V,E),drs([],[rel(arg2,E,X)]))))) :
    ((4_v -o 4_t) -o (2_e -o (4_v -o 4_t))) || noscope
lam(X,drs([],[eq(X,'Susan')])) : (8_e -o 8_t)
lam(V,lam(X,lam(E,merge(app(V,E),drs([],[rel(arg1,E,X)]))))) :
    ((4_v -o 4_t) -o (6_e -o (4_v -o 4_t))) || noscope
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X))))) :
    ((8_e -o 8_t) -o ((8_e -o 3_t) -o 3_t)) || noscope
lam(V,drs([],[pred(give,V)])) : (4_v -o 4_t)
lam(X,drs([],[eq(X,'Mary')])) : (6_e -o 6_t)
}
```

F-structure:

```
"Susan was given the bear by Mary"

[PRED      'give<[87:Mary], [53:bear], [1:Susan]>'                          ]
[          1[PRED 'Susan'                                                   ]]
[SUBJ    125[CASE nom, GEND fem, NTYPE name, NUM sg, PERS 3                 ]]
[        127[                                                               ]]
[          72[PRED 'bear'                                                   ]]
[         163[DEF +, NTYPE count, NUM sg, PERS 3                            ]]
[OBJ2      53[                                                              ]]
[          71[                                                              ]]
[         165[                                                              ]]
[          96[PRED 'Mary'                                                   ]]
[         177[GEND fem, NTYPE name, NUM sg, PERS 3, PFORM by, PTYPE nosem   ]]
 26       179[                                                              ]
139 OBL-AG  87[                                                              ]
 14         95[                                                              ]
 15        183[                                                              ]]
188
202 TNS-ASP   [TENSE past                                                    ]
204 [DATIVE-SHIFT +, PARTICIPLE past, PASSIVE +                              ]
```

(19) Mary hugged herself.



Produced meaning constructors:

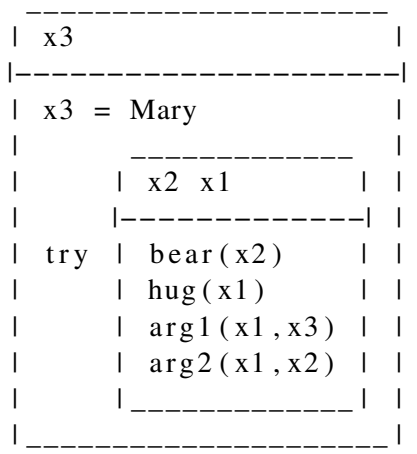
```
{
lam(X,drs([],[eq(X,'Mary')])) : (6_e -o 6_t)
lam(A,alfa(B,refl,pred(female,B),merge(app(A,C),drs([C],
    [pred(female,C),eq(B,C)])))) : ((2_e -o 3_t) -o 3_t)
lam(V,lam(X,lam(E,merge(app(V,E),drs([],[rel(arg1,E,X)]))))) :
    ((4_v -o 4_t) -o (6_e -o (4_v -o 4_t))) || noscope
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X))))) :
    ((6_e -o 6_t) -o ((6_e -o 3_t) -o 3_t))
lam(V,drs([],[pred(hug,V)])) : (4_v -o 4_t)
lam(V,merge(drs([E],[]),app(V,E))) : ((4_v -o 4_t) -o 3_t)
lam(V,lam(X,lam(E,merge(app(V,E),drs([],[rel(arg2,E,X)]))))) :
    ((4_v -o 4_t) -o (2_e -o (4_v -o 4_t))) || noscope
}
```

F-structure:

```
"Mary hugged herself"

      [PRED    'hug<[1:Mary], [23:herself]>'                          ]
      [       1[PRED 'Mary'                                           ]
      [SUBJ  61[CASE nom, GEND fem, NTYPE name, NUM sg, PERS 3        ]
      [      63[                                                      ]
      [       23[PRED 'herself'                                       ]
   14 [OBJ    24[CASE acc, NTYPE pron, NUM sg, PERS 3, PRON-TYPE pers ]
   73 [       86[                                                     ]
   88 [
  102 [TNS-ASP [MOOD indicative, PERF -_, PROG -_, TENSE past]        ]
  105 [PASSIVE -                                                      ]
```

(20) Mary tried to hug a bear.



Produced meaning constructors:

```
{
lam(V,lam(X,lam(E,merge(app(V,E),drs([],[rel(arg2,E,X)])))))  :
    ((11_v -o 11_t) -o (9_e -o (11_v -o 11_t)))  ||  noscope
lam(X,drs([],[pred(bear,X)]))  :  (9_e -o 9_t)
lam(X,drs([],[eq(X,'Mary')]))  :  (2_e -o 2_t)
lam(V,lam(X,lam(E,merge(app(V,E),drs([],[rel(arg1,E,X)])))))  :  (
    (11_v -o 11_t) -o (2_e -o (11_v -o 11_t)))  ||  noscope
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X)))))  :
    ((2_e -o 2_t) -o ((2_e -o 3_t) -o 3_t))  ||  noscope
lam(V,drs([],[pred(hug,V)]))  :  (11_v -o 11_t)
lam(X,lam(P,drs([],[try(app(P,X))])))  :  (2_e -o ((2_e -o 10_t) -o 3_t))
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X)))))  :
    ((9_e -o 9_t) -o ((9_e -o 10_t) -o 10_t))
lam(V,merge(drs([E],[]),app(V,E)))  :  ((11_v -o 11_t) -o 10_t)
}
```

F-structure:

```
"Mary tried to hug a bear"

 [PRED     'try<[1:Mary], [29:hug]>'                                    ]
 [        1[PRED 'Mary'                                                 ]
 [SUBJ   95[CASE nom, GEND fem, NTYPE name, NUM sg, PERS 3              ]
 [       97[                                                            ]
 [          [PRED   'hug<[1:Mary], [55:bear]>'                          ]
 [          [SUBJ   [1:Mary]                                            ]
 [       39[     64[PRED 'bear'                                         ]
 [XCOMP 124[    148[                                                    ]
 [      154[OBJ  55[SPEC [DET [PRED 'a']]                              ]
 [       29[     63[CASE acc, DEF -, NTYPE count, NUM sg, PERS 3        ]
  14      37[    150[                                                    ]
 107     168[PASSIVE -, VFORM inf                                       ]
 170
 173[TNS-ASP  [MOOD indicative, PERF -_, PROG -_, TENSE past]           ]
 175[VFORM    inf                                                       ]
```

(21) Mary saw the bear with the telescope

```
 _____________________                 _____________________
| x2 x3 x4 x1         |               | x3 x2 x4 x1         |
|---------------------|               |---------------------|
| bear(x2)            |               | bear(x3)            |
| x3 = Mary           |               | x2 = Mary           |
| telescope(x4)       |               | telescope(x4)       |
| with(x1,x4)         |               | with(x3,x4)         |
| see(x1)             |               | see(x1)             |
| arg1(x1,x3)         |               | arg2(x1,x3)         |
| arg2(x1,x2)         |               | arg1(x1,x2)         |
|_____________________|               |_____________________|
```

Produced meaning constructors:

```
{
lam(X,drs([],[pred(telescope,X)])) : (4_e -o 4_t)
lam(V,merge(drs([E],[]),app(V,E))) : ((6_v -o 6_t) -o 5_t)
lam(V,lam(X,lam(E,merge(app(V,E),drs([],[rel(arg2,E,X)]))))) :
    ((6_v -o 6_t) -o (9_e -o (6_v -o 6_t))) || noscope
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X))))) :
    ((4_e -o 4_t) -o ((4_e -o 5_t) -o 5_t)) || noscope
lam(X,drs([],[eq(X,'Mary')])) : (11_e -o 11_t)
lam(V,lam(X,lam(E,merge(app(V,E),drs([],[rel(arg1,E,X)]))))) :
    ((6_v -o 6_t) -o (11_e -o (6_v -o 6_t))) || noscope
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X))))) :
    ((11_e -o 11_t) -o ((11_e -o 5_t) -o 5_t)) || noscope
lam(X,drs([],[pred(bear,X)])) : (9_e -o 9_t)
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X))))) :
    ((9_e -o 9_t) -o ((9_e -o 5_t) -o 5_t)) || noscope
lam(Y,lam(X,drs([],[rel(with,X,Y)]))) :
    (4_e -o (6_v -o 7_t))
lam(U,lam(V,lam(E,merge(drs([],[]),merge(app(U,E),app(V,E)))))) :
    ((6_v -o 7_t) -o ((6_v -o 6_t) -o (6_v -o 6_t)))
lam(V,drs([],[pred(see,V)])) : (6_v -o 6_t)
}

{
lam(X,drs([],[pred(telescope,X)])) : (5_e -o 5_t)
lam(V,merge(drs([E],[]),app(V,E))) : ((7_v -o 7_t) -o 6_t)
lam(U,lam(V,lam(E,merge(drs([],[]),merge(app(U,E),app(V,E)))))) :
    ((9_e -o 8_t) -o ((9_e -o 6_t) -o (9_e -o 6_t))) || noscope
lam(V,lam(X,lam(E,merge(app(V,E),drs([],[rel(arg2,E,X)]))))) :
    ((7_v -o 7_t) -o (9_e -o (7_v -o 7_t))) || noscope
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X))))) :
    ((5_e -o 5_t) -o ((5_e -o 6_t) -o 6_t)) || noscope
lam(X,drs([],[eq(X,'Mary')])) : (11_e -o 11_t)
lam(V,lam(X,lam(E,merge(app(V,E),drs([],[rel(arg1,E,X)]))))) :
    ((7_v -o 7_t) -o (11_e -o (7_v -o 7_t))) || noscope
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X))))) :
    ((11_e -o 11_t) -o ((11_e -o 6_t) -o 6_t)) || noscope
lam(X,drs([],[pred(bear,X)])) : (9_e -o 9_t)
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X))))) :
    ((9_e -o 9_t) -o ((9_e -o 6_t) -o 6_t)) || noscope
lam(Y,lam(X,drs([],[rel(with,X,Y)]))) : (5_e -o (9_e -o 8_t))
lam(V,drs([],[pred(see,V)])) : (7_v -o 7_t)
}
```

F-structures:

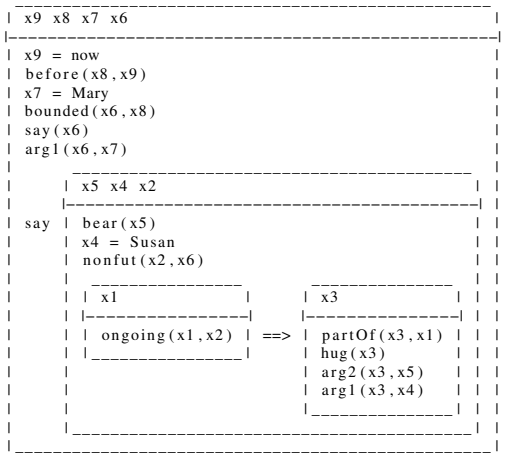
"Mary saw the bear with the telescope"

```
 [PRED    'see<[1:Mary], [37:bear]>'
       1 [PRED 'Mary'
SUBJ 136 [CASE nom, GEND fem, NTYPE name, NUM sg, PERS 3]
     138]
      56 [PRED 'bear'
     172 [CASE acc, DEF +, NTYPE count, NUM sg, PERS 3]
OBJ   37
      55
     174]
          { [PRED   'with<[86:telescope]>'
             105 [PRED 'telescope'
             193 [CASE acc, DEF +, NTYPE count, NUM sg, PERS 3]
ADJUNCT {  OBJ  86
          71  104
  14      85  195]
 148     199 PTYPE  sem]}
 204
 218 TNS-ASP [MOOD indicative, PERF -_, PROG -_, TENSE past]
 220 PASSIVE -]
```

"Mary saw the bear with the telescope"

```
 [PRED    'see<[1:Mary], [37:bear]>'
       1 [PRED 'Mary'
SUBJ 136 [CASE nom, GEND fem, NTYPE name, NUM sg, PERS 3]
     138]
          [PRED    'bear'
                  { [PRED   'with<[86:telescope]>'
                     105 [PRED 'telescope'
                     193 [CASE acc, DEF +, NTYPE count, NUM sg, PERS 3]
OBJ      56 ADJUNCT {  OBJ  86
          71         104
     172  85         195]
      37 199 PTYPE  sem]}
  14  55
 148 201 CASE acc, DEF +, NTYPE count, NUM sg, PERS 3]
 204
 218 TNS-ASP [MOOD indicative, PERF -_, PROG -_, TENSE past]
 220 PASSIVE -]
```

(22) Mary said that Susan was hugging a bear.



Produced meaning constructors:

```
{
//Liger
lam(S,lam(T,drs([],[rel(ongoing,T,S)]))) : (207_s -o (209_s -o 205_t))
lam(S,lam(T,drs([],[rel(bounded,T,S)]))) : (208_s -o (210_s -o 206_t))
lam(M,lam(P,lam(S,drs([],[imp(merge(drs([Z],[]),app(app(M,S),Z)),app(P,Z))])))) :
    ((207_s -o (209_s -o 205_t)) -o ((10_s -o 6_t) -o (11_s -o 6_t)))
lam(M,lam(P,lam(S,merge(drs([Z],[]),merge(app(app(M,S),Z),app(P,Z)))))) :
    ((208_s -o (210_s -o 206_t)) -o ((19_s -o 18_t) -o (8_s -o 18_t)))
lam(T,lam(T2,drs([],[rel(before,T,T2)]))) : (8_s -o (9_s -o 8_t))
lam(T,lam(T2,drs([],[rel(nonfut,T,T2)]))) : (11_s -o (12_s -o 11_t))
lam(T,lam(P,lam(S,merge(drs([R],[]),merge(app(app(T,R),S),app(P,R)))))) :
    ((11_s -o (12_s -o 11_t)) -o ((11_s -o 6_t) -o (12_s -o 6_t)))
lam(T,lam(P,lam(S,merge(drs([R],[]),merge(app(app(T,R),S),app(P,R)))))) :
    ((8_s -o (9_s -o 8_t)) -o ((8_s -o 18_t) -o (9_s -o 18_t)))
//Grammar
lam(X,drs([],[eq(X,'Susan')])) : (14_e -o 14_t)
lam(X,drs([],[eq(X,'Mary')])) : (17_e -o 17_t)
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X))))) :
    ((5_e -o 5_t) -o ((5_e -o 6_t) -o 6_t))
lam(X,drs([],[pred(bear,X)])) : (5_e -o 5_t)
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X))))) :
    ((17_e -o 17_t) -o ((17_e -o 18_t) -o 18_t)) || noscope
lam(V,lam(X,lam(E,merge(app(V,E),drs([],[rel(arg2,E,X)]))))) :
    ((7_v -o 7_t) -o (5_e -o (7_v -o 7_t))) || noscope
lam(P,merge(drs([T],[eq(T,now)]),app(P,T))) :
    ((9_s -o 18_t) -o 18_t) || noscope
lam(P,lam(Q,merge(drs([X],[]),merge(app(P,X),app(Q,X))))) :
    ((14_e -o 14_t) -o ((14_e -o 6_t) -o 6_t)) || noscope
lam(V,lam(X,lam(E,merge(app(V,E),drs([],[rel(arg1,E,X)]))))) :
    ((7_v -o 7_t) -o (14_e -o (7_v -o 7_t))) || noscope
lam(V,drs([],[pred(hug,V)])) : (7_v -o 7_t)
lam(P,lam(X,lam(S,merge(drs([],[pred(say,S),rel(arg1,S,X)]),drs([],[say(app(P,S))]))))) :
    ((12_s -o 6_t) -o (17_e -o (19_s -o 18_t)))
lam(V,lam(S,merge(drs([E],[rel(partOf,E,S)]),app(V,E)))) :
    ((7_v -o 7_t) -o (10_s -o 6_t))
}
```

F-structure:

"Mary said that Susan hugged a bear"

```
[PRED    'say<[1:Mary], [23:hug]>'
      1 [PRED 'Mary'                                                    ]
SUBJ 123[CASE nom, GEND fem, NTYPE name, NUM sg, PERS 3                 ]
     125[                                                               ]
        [PRED    'hug<[58:Susan], [83:bear]>'                           ]
          58[PRED 'Susan'                                               ]
SUBJ     155[CASE nom, GEND fem, NTYPE name, NUM sg, PERS 3             ]
         157[                                                           ]
           92[PRED 'bear'                                               ]
COMP   71 191[                                                          ]
      167OBJ  83[SPEC [DET [PRED 'a']]                                  ]
      197      91[CASE acc, DEF -, NTYPE count, NUM sg, PERS 3          ]
      211     193[                                                      ]
       23
  14   57[TNS-ASP [MOOD indicative, PERF -_, PROG -_, TENSE past]       ]
 135  213[COMP-FORM that, PASSIVE -                                      ]
 215
 218[TNS-ASP [MOOD indicative, PERF -_, PROG -_, TENSE past]
 220[ROOT    +                                                          ]
```

Which Model Mimics Human Mental Lexicon Better? A Comparative Study of Word Embedding and Generative Models

Huacheng Song Zhaoxin Feng Emmanuele Chersoni Chu-Ren Huang
Department of Language Science and Technology, The Hong Kong Polytechnic University
{huacheng.song, zhaoxinbetty.feng}@connect.polyu.hk
{emmanuele.chersoni, churen.huang}@polyu.edu.hk

Abstract

Word associations are commonly applied in psycholinguistics to investigate the nature and structure of the human mental lexicon, and at the same time an important data source for measuring the alignment of language models with human semantic representations.

Taking this view, we compare the capacities of different language models to model collective human association norms via five word association tasks (WATs), with predictions about associations driven by either word vector similarities for traditional embedding models or prompting large language models (LLMs).

Our results demonstrate that neither approach could produce human-like performances in all five WATs. Hence, none of them can successfully model the human mental lexicon yet. Our detailed analysis shows that static word-type embeddings and prompted LLMs have overall better alignment with human norms compared to word-token embeddings from pre-trained models like BERT. Further analysis suggests that the performance discrepancies may be due to different model architectures, especially in terms of approximating human-like associative reasoning through either semantic similarity or relatedness evaluation[1].

1 Introduction

Artificial intelligence, particularly large language models (LLMs), functionally emulates the way we humans perceive and conceptualize the physical reality, as well as how we understand and process multifaceted information (Löhn et al., 2024). Yet a pivotal open question remains unsolved: to what extent do LLMs align with the conceptual knowledge hierarchically encoded in human cognition as their capabilities advance? This is where the "machine psychology" comes into play to scrutinize LLMs' "behavioral traits" and "thinking patterns" through

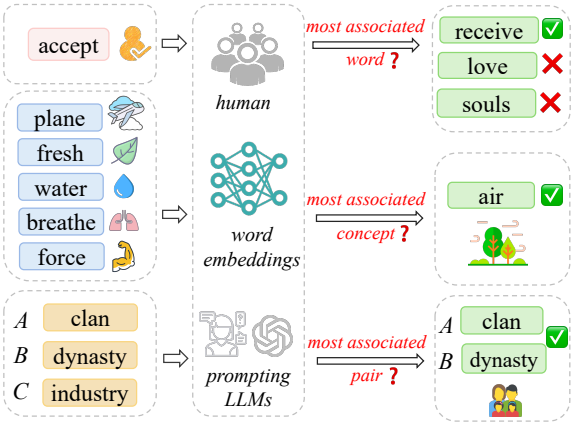


Figure 1: Illustration of Common Word Association Tasks. These tasks evaluate semantic alignment between computational models (word embeddings vs. LLM prompting) and human-like associative reasoning.

psychological tests adapted from interpretable research on human (Hagendorff, 2023).

Successful modeling of the human mental lexicon can be viewed as an essential step in verifying human-like intelligence. Human mental lexicon, in contrast to electronic lexica, is extremely versatile in supporting the association and generation of new concepts. Indeed *word association norms* is a typical method of investigation: a stimulus word is presented to a human participant, who is simply required to produce the first word coming to mind (McRae et al., 2012; De Deyne et al., 2019). Semantic similarities and relatedness that underlie the core of human mental lexicon is hereby quantified as collective linguistic norms. Since distributional similarity between words is an important factor explaining associations, traditional studies extensively adopted Distributional Semantic Models (DSMs) and word embeddings to predict human word associations (Mandera et al., 2017; Evert and Lapesa, 2021; Kwong et al., 2022; A et al., 2024). On the other hand more recent studies, based on LLMs, proved that such systems can align, to a considerable extent, with human patterns of asso-

[1]Our codes and data are publicly available at https://github.com/florethsong/word_association

Proceedings of the 16th International Conference on Computational Semantics, pages 218–240
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

ciating words (De Deyne et al., 2024; Abramski et al., 2025; Bai et al., 2025). An open question arisen is which one of these methods delivers better results in approximating human norms.

Besides the theoretical interest of the problem, the results are relevant to the problem of *reverse dictionary*, where a user tries to retrieve a word given a set of associates or a dictionary definition (Almeman and Espinosa-Anke, 2024). Reverse dictionary applications, which can be seen as the information retrieval modeling side of human lexical access (the so-called *tip-of-the-tongue*, *anomia* or *dysnomia* problem, see Zock (2002) and Rapp and Zock (2014)) can be helpful tools for writers and translators, and in this sense, generative LLMs show a lot of promise, as they could help a user by retrieving and generating a target word simply on the basis of a prompt with some lexical cues. From a psychological standpoint, word associations are also a fundamental indicator for human *creativity* and *divergent thinking*, as research indicates a consistent positive correlation between high levels of human creativity and the capacity to generate word associates that are distant in the lexical network (Kenett and Faust, 2019; Yang et al., 2022; Johnson and Hass, 2022; Wang et al., 2024).

As illustrated by the task types in Figure 1 focusing on semantic similarity and relatedness, this study designs a protocol of five-stage word association tasks (WATs) to evaluate models against human norms. By taking the majority of human responses across various WATs as a main proxy of human mental lexicon, this study compare the word association abilities of vectors from traditional static word-type embedding models (WEMs), mean-pooled word-token embeddings from representative pretrained language models (PLMs), and prompting strategies with mainstream LLMs. Results show that although none of these models align fully with human mental lexicon and hence model effectively the versatility of the human cognitive ability, WEMs and LLMs can better mimic human associations than PLMs: LLMs outperform competitors in word retrieval tasks (with a focus on capturing semantic similarities, i.e., lexical interchangeability), while WEMs perform better in concept pairing (emphasizing the identification of semantic relatedness, that is, detecting mutual conceptual relations). While scaling-up and contextualization often helps embedding models, PLMs show more architecture- and task-dependent trade-offs.

2 Related Work

WATs with Humans Word associations are grounded in Firthian's "word in company" tradition that lexemes with resembling behavioral profiles (like, shared collocational patterns or syntactic structures) encode similar paradigmatic or syntagmatic relations in meaning and cognition (Firth, 1957; Church and Hanks, 1990). They function as prototypical and advantageous tools in psycholinguistics to tap directly into semantic memory and conceptual knowledge reflected in human thinking, reasoning, and language use. As a classical paradigm, the free word association task and its variants based on word clustering or relationship identification accelerate quantitative exploration of human cognitive phenomena, such as language acquisition (Citraro et al., 2023), metaphor and analogy comprehension (Lu et al., 2022), and creativity (Beaty and Kenett, 2023; Wang et al., 2024).

Various human association norms originally designed to access preexisting word knowledge in the human mind and detect different aspects of cognitive development and competencies, such as EAT (the Edinburgh Associative Thesaurus, Kiss et al., 1973), USF (the University of South Florida Free Association Norms, Nelson et al., 2004), and SWOW (the Small World of Words, De Deyne et al., 2019), can be applied in conjunction as a comprehensive benchmark for facilitating the measurement of the alignment between human internal semantic cognition and external word embeddings.

WATs with Word Embeddings WATs have significantly contributed to benchmarking models' semantic representations and conceptual structures against human mental lexicon shown in diverse human-generated norms, both in theory and practice (Rapp and Zock, 2014; De Deyne et al., 2016). They provide a powerful means to probe into two fundamental dimensions of distributional semantics: *similarity* (interchangeability of words, e.g., *car/van*) and *relatedness* (shared conceptual relations between words, e.g., *car/wheel*) (Fodor et al., 2023). Existing work (Lenci et al., 2022; Fodor et al., 2023; A et al., 2024, etc.) has been extensively devoted to thorough comparisons across a wide spectrum of DSMs from count (e.g., Dissect PPMI, Baroni et al., 2014) and predict models (e.g., word2vec, Mikolov et al., 2013) at early static-embedding generation to recent transformer-based contextual embedding models (e.g., BERT, Devlin

et al., 2019). These studies consistently demonstrated the superior performance of static embeddings in out-of-context WATs, while highlighting contextual embeddings' advantages in tasks requiring contextual sensitivity. Collectively, they revealed the nuanced interplay between model design, task requirements, and cognitive plausibility of language representations.

WATs with LLMs Recent work expanded the use of WATs into dissecting the behaviors of LLMs as black-box systems to better understand their advantages and limitations in semantic-aware reasoning. Abramski et al. (2025) established LLM-generated free association norms by prompting popular LLMs and found that LLM-generated associations exhibit weaker concreteness effects and stronger societal biases compared to human norms. Cazalets and Dambre (2025) demonstrated GPT-series' ability to synchronize with human players in game-like free association interactions. Beyond free association tasks, structured variants such as ontological classification (De Deyne et al., 2024), connection tasks (Samdarshi et al., 2024), and similarity judgments on triads (Linhardt et al., 2025) have assessed LLMs' ability to identify underlying internal relations or cluster words by shared characteristics. Increasing interest has been in using WATs to reveal both explicit and implicit societal biases encoded in LLMs. For example, studies by Ethayarajh et al. (2019), Abramski et al. (2025), and Bai et al. (2025) presented how WATs can uncover attitude disparities between model outputs and human responses, highlighting their utility in addressing ethical issues of language models.

Such studies stress WATs' dual role in illuminating human and models' semantic networks; however, existing work mainly relied either on prompt-based strategies with LLMs or on embedding similarity, without any systematic comparison between the two. Also, previous studies were limited in scope, focusing only on one type of WAT, therefore a more comprehensive evaluation is necessary.

3 Experimental Settings

According to Abramski et al. (2025), probing into the conceptual knowledge encoded within language models by examining the embedding space works well for traditional models, but it is less effective and practical for LLMs. This is due to the fact that embeddings from LLMs exhibit severe anisotropy in their vector spaces, which can significantly distort similarity estimates (e.g., Ethayarajh, 2019; Zhang et al., 2020; Biś et al., 2021; Timkey and van Schijndel, 2021; Nie et al., 2025; Feng et al., 2025). Therefore, a shift from the conventional approach of accessing the embedding space to a top-down approach in the context of LLMs was proposed, which means directly prompting LLMs with specific tasks and using their outputs to infer the knowledge in their vector spaces.

Therefore, we examine the capabilities of different models by employing two methodologies: **embedding** and **prompting**, which align with their default typical approaches to WATs at hand. A basic assumption of embedding-based tests is that *the strength of word associations increases with the cosine similarity of their embeddings* (Clark, 2015; Fodor et al., 2023), reflecting graded semantic relationships in vector spaces. For WEMs, we extracted static word-type embeddings and calculated the cosine similarities as the basis for their outputs. In terms of PLMs, both non-contextualized and contextualized word embeddings were mean-pooled from the last hidden layers and cosine similarities were computed. Regarding LLMs, we utilized zero-shot prompts to obtain direct responses.

3.1 Task Design

We tested our models on five complementary and progressively challenging tasks built on the well-established datasets, as summarized in Table 1. Each task stresses distinct capabilities of language models in terms of processing semantic similarity versus relatedness, with extended discussion provided in Appendix A.

Task 1: Multiple-Choice Associations FAST dataset (Evert and Lapesa, 2021) is leveraged in this task, which provides quadruples of a stimulus and three candidate words: "*FIRST*, *HAPAX*, *RANDOM*" where *FIRST* is the most frequent associate response from humans, *HAPAX* is a response that has been mentioned only once, and *RANDOM* is a randomly selected control candidate with minimal semantic association strength to the stimulus. For each stimulus, a model has to choose the most strongly associated word (i.e., for embedding models, the one with the largest semantic similarity). It is worth noticing that *HAPAX* is also a word with weak semantic association with the stimulus, and thus it works as a strong distractor.

Performance is measured using *Accuracy*, i.e., the percentage of items in which the model cor-

Table 1: Overview of Datasets for the Five Association Tasks. In the "Structure" column, underlined elements indicate the information presented to the evaluated models, while bolded elements are used as the ground truth.

Task	Dataset	Structure	Size[2]	Word List	Metrics[3]
1	FAST	<stimulus, ***FIRST***, *HAPAX*, *RANDOM*> *(e.g., accept,* ***receive****, love, souls)*	11,431 (12,329)		Accuracy
2	FAST	<stimulus, ***FIRST***, *HAPAX*, *RANDOM*> *(e.g., achievement,* ***success****, degree, round)*	11,431 (12,329)	✓	Top-1 Accuracy, Mean Rank (threshold = 4)
3	CogALex	<**Target**, a1, a2, a3, a4, a5> *(e.g.,* ***air****, plane, fresh, water, breathe, force)*	3,650 (4,000)	✓	Top-1 Accuracy, Mean Rank (threshold = 4)
4	Concrete-Abstract Triad	<A, B, C> ($P_{\mathbf{AB}}$, $P_{\mathbf{AC}}$, $P_{\mathbf{BC}}$) *(concrete e.g., banana, cherry, pineapple* ***(0.18, 0.65, 0.18)****)* *(abstract e.g., darling, hero, thinker* ***(0.48, 0.13, 0.40)****)*	100 + 100		Accuracy (Total, Concrete, Abstract)
5	Remote Triad	<A, B, C> ($P_{\mathbf{AB}}$, $P_{\mathbf{AC}}$, $P_{\mathbf{BC}}$) *(e.g., fence, mask, salt* ***(0.80, 0.05, 0.15)****)*	100		Accuracy

rectly picks the *FIRST* associate ($[0\%, 100\%]$), with a random-choice baseline of 33.3%. To mitigate potential positional bias, the elements in each candidate list were shuffled during LLM prompting.

Task 2: Open-Vocabulary Associations This task also relies on the FAST dataset but differs in that it presents no fixed set of candidates. Instead, models are asked to generate the most associated word in an open-vocabulary setup which further simulates the way humans access their mental lexicon in a natural association task.

In the current study, we create a "pseudo-open vocabulary" condition for WEMs and PLMs where models are tasked with ranking associations for a given stimulus over a large-scale word list, which covers all *FIRST* words and restricts the range of potential choices. The tailored word list applied in this study is a concatenation of vocabularies from word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and FastText (Joulin et al., 2017) models, totaling 101,607 word types, effectively serving the goals of the task. While LLMs are asked to directly provide 30 words associated with the stimulus, ordered by their association strength.

Two statistical metrics are reported based on the word ranking list for each stimulus (i.e., the word list sorted by decreasing cosine similarity based on embedding-based models, and the ranked word list generated by LLMs): 1) *Top-1 Accuracy*: how frequently a model ranks the *FIRST* human response as the top 1 result ($[0\%, 100\%]$), positively correlated with model-human semantic alignment; and 2) *Mean Rank (threshold = 4)*: the average position of the *FIRST* word in the rankings by a certain model. We set *4* as the threshold, that is, if the rank of the *FIRST* word in a given ranking list is *3* or lower, we assign this actual rank as the score for the given instance, otherwise we assign a score of *4*. This is in line with the convention of shared tasks using mean rank to mitigate excessive penalty on instances with high-rank outliers (Camacho-Collados et al., 2018; Mansar et al., 2021). The final scores are mean ranks falling in $[1, 4]$, which are negatively correlated with the performance of models in lexical alignment with humans.

Task 3: Reverse Associations Based on the CogALex shared task dataset (Rapp and Zock, 2014), this task evaluates the models' ability to simultaneously integrate multi-layered relations across multiple stimuli. The logic of this task is closely related to the *tip-of-the-tongue* phenomenon. Each item features a *Target* word defined as the human-generated response to five given cue words, which are all interconnected with the *Target* at a certain conceptual level.

The objective is to retrieve the *Target* word that semantically connects the five cue words, within a pseudo-open vocabulary of candidates. For WEMs and PLMs, we compute the average vector of the five cue words and measure the association strength (i.e., cosine similarity) between it and each candi-

[2]Since word2vec, GloVe, and FastText models underperform when faced with out-of-vocabulary words, we manually excluded any missing items if a word in our specific word set is not included in any of these three baseline models. As a result, we obtained 11,431 out of 12,329 items from the original FAST dataset for Tasks 1 and 2, and 3,650 out of 4,000 items from the original CogALex dataset for Task 3. For both triad datasets corresponding to Tasks 4 and 5, no items were removed from the original datasets.

[3]To ensure reliable and effective comparisons, we conducted two types of significance tests, depending on the evaluation metrics. For accuracy scores in Tasks 1–5, we applied McNemar's test (McNemar, 1947) corrected with Benjamini–Hochberg procedure (Benjamini and Hochberg, 1995) across all model pairs to determine whether the observed accuracy differences are statistically significant. For mean rank results in Tasks 2 and 3, we used the Wilcoxon signed-rank test (Wilcoxon, 1945) to evaluate whether the rankings of the *FIRST* or *Target* words in the given instances produced by different models differed significantly. More details can be found in Figures 8-12 in the Appendices.

date word in a list of 101,607 words (identical to that used in Task 2) to produce a ranked list of target words, while LLMs are required to directly generate a list of 30 potential targets. Performance is evaluated using the same two metrics as in Task 2. This task emphasizes reverse reasoning ability and tests whether models can reconstruct a unifying concept from distributed cues.

Task 4: Concrete-Abstract Association Triads This task presents triads of words to models, where any two can be paired based on varying semantic features. The goal is to select the most semantically related pair in each triad. The dataset, introduced by De Deyne et al. (2021) is employed, which can be split into two subsets: 1) **Concrete Triad Dataset** focusing on physical entities and events; 2) **Abstract Triad Dataset** focusing on psychological and conceptual relationships.

Models' outputs are compared against human preferences with percentages provided in the dataset. Specifically, for each instance, WEMs and PLMs select the word pair with the highest cosine similarity among the three candidate pairs based on their word embeddings, whereas the top-ranked pair from all three pairs is regarded as LLMs' final choice. We report respectively the accuracies on total, concrete, and abstract triads, all ranging in $[0\%, 100\%]$ and positively correlated with model-human alignment. In cases where humans do not produce a single dominant pairing (e.g., two pairings have equal frequencies chosen by humans), a model's choice is considered correct if it matches one of the most frequent human choices.

Task 5: Remote Association Triads Similar to the structure in Task 4 but significantly more challenging, this task utilizes the Remote Triad dataset (De Deyne et al., 2016) and requests models to identify the most related pairing with more distant and creative semantic links among words. As in Task 4, we measure accuracy based on human preferences provided in the original dataset. Due to the subtlety of the associations involved, this task offers deeper and informative insights into the extent to which models can capture latent and implicit conceptual relations beyond immediate meaning similarity between words.

3.2 Model Selection

We evaluate representative and state-of-the-art language models across three architectural paradigms and development stages, further dividing them into "*Smaller*" (with around 1B or fewer parameters) and "*Larger*" (with over 1B parameters) categories based on parameter scale. No post hoc modifications were conducted to the vanilla models and their embeddings with the intention to assess the intrinsic quality of their representations.

The first group covers five static **WEMs**: *word2vec* (Mikolov et al., 2013) pretrained on 100B tokens of Google News, *GloVe* (Pennington et al., 2014) trained on 6B tokens of Wikipedia 2014 and newspapers as well as *GloVe-CC* on 840B tokens of Common Crawl (CC) Web data, and *FastText* (Joulin et al., 2017) trained on 16B tokens of Wikipedia 2017 and other webbase corpus as well as *FastText-CC* on 600B tokens of CC. All models were tested with 300-dimensional embeddings.

The second group includes six **PLMs**: *BERT-base* and *-large* (Devlin et al., 2019), *GPT-2* and *-xl* (Radford et al., 2019), and *T5-small* and *-3B* (Raffel et al., 2020), from which we extracted non-contextualized (the input is a single word, like "*accept*") as well as contextualized (the input is a fixed simple sentence containing the key word, like "*My target word is accept*") word embeddings by mean-pooling the subword representations in the last layers.

The third group composes three **LLMs**, i.e., *GPT-4.1*[3], *DeepSeek-V3 (-0324)* (DeepSeek-AI, 2024), and *Qwen3 (-238B-A22B)* (Yang et al., 2025). We ran additional experiments (cf. Appendix G) to test how different temperature settings (0.01 vs. 0.5 vs. 1), prompt strategies (simple zero-shot vs. enhanced few-shot), and reasoning modes (standard vs. reasoning) impact LLM effectiveness across different WATs. While results indicate that most LLMs achieve marginally better performance at temperature 0.5 using detailed few-shot prompts with reasoning, optimal configurations vary across tasks and models. To obtain consistent and comparable patterns from LLMs, we standardized our configurations: temperature was maintained at 0.01 using zero-shot prompts, and reasoning ability was not activated for the reasoning model—Qwen3.

4 Results and Analysis

This section reports the empirical results and findings obtained from operationalizing the series of tasks and metrics defined in Section 3.1. The statistics corresponding to each task and significance test results are displayed in Appendices B-F.

[2]https://openai.com/index/gpt-4-1/

4.1 Multiple-Choice Association

Figure 2 illustrates the performance of various language models in Task 1, that is, identifying the most interchangeable word or near-synonyms to a given cue from a restricted set of candidates. With the only exception of GPT-2 (non-ctx), all models achieve an accuracy vastly better than the chance-level baseline. Notably, WEMs and LLMs significantly outperform PLMs proved by Figure 8, frequently reaching accuracies of 80% or higher. This suggests that static word-type representations derived from WEMs and prompted LLMs are more effective at capturing direct semantic similarities between near-synonyms or conceptually related words. In contrast, token-level embeddings mean-pooled from PLMs show substantially reduced effectiveness, indicating a difficulty in abstracting a type-level representation, which would be necessary for this task. Our findings are consistent with Lenci et al. (2022), Apidianaki (2023), and A et al. (2024), who claimed that word-token representations complicate the investigation of lexical semantic knowledge anchored at the word-type level.

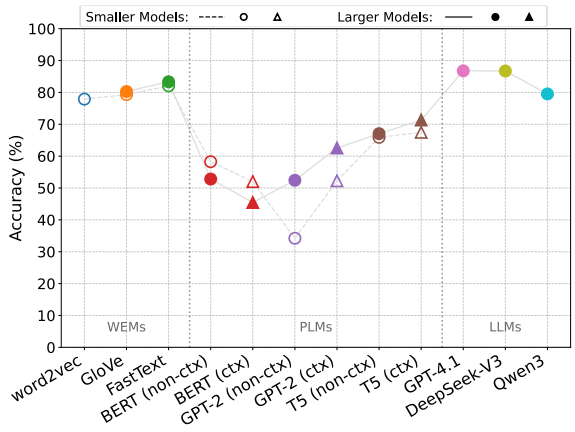


Figure 2: Plot of Model Accuracies in the Multiple-Choice Association Task. Fillings and shapes are used to distinguish the context types and the magnitudes of models. Hollow markers indicate smaller models, while solid ones represent larger ones. Non-contextualized (non-ctx) PLMs are shown as circles, in contrast to contextualized (ctx) PLMs marked with triangles. *Note: the visual markers in the subsequent figures maintain consistent meanings throughout this paper.*

Additionally, when comparing the efficiency of non-contextualized embeddings to contextualized ones within PLMs, it is interesting to note that extra contexts benefit both GPT-2 and T5, though to varying degrees, while BERT-base and BERT-large models do not display the same enhancement.

Comparisons between smaller and larger models reveal that, for most WEMs and PLMs, increasing parameter count correlates with improved modeling of lexical semantics and conceptual relationships. Larger models tend to outperform smaller ones, aligning with established *Scaling Laws* (Kaplan et al., 2020), with the exception of BERT, whose larger variant is worse than the smaller one, pointing to its potential architectural or training-related limitations in preserving word-type knowledge during scaling-up.

Figure 3 reveals distinct error patterns across different model types. The errors align with overall accuracy trends: WEMs and LLMs predominantly select *HAPAX*, indicating a relatively strong sensitivity to weak associations, while making few *RANDOM* selections. This suggests that such models can at least effectively distinguish between weak and non-existent associations, while in contrast PLMs and particularly GPT-2 (non-ctx) are more frequently misled by *RANDOM* distractors. Furthermore, LLMs occasionally encountered *OTHER* errors, particularly involving incorrect formats or range misinterpretations under zero-shot prompting. For example, LLMs may output *stock* in response to *garters* with the candidate list [*lace*, *sweaters*, *stockings*], reflecting possible failures in instruction following that manifest as hallucinations or misalignment with task requirements.

4.2 Open-Vocabulary Association

Task 2 introduces a more demanding evaluation scenario, placing models under empirically unrestricted "free" association conditions, therefore resulting in universally lower performance across all models as evidenced in Figure 4. This task probes the models' global semantic organization and broader vector space in that they mirror human-like associative knowledge. Remarkably, the stark disparities in top-1 accuracies and mean ranks between WEMs/PLMs and LLMs (the majority of these differences are statistically significant as shown in Figure 9) highlight that LLMs can more reliably identify human-preferred associative targets by frequently retrieving and prioritizing near-synonyms of high-frequency co-occurring lexemes for the given stimulus (e.g., *really* for *actually*, *departure* for *arrival*).

Interestingly, the effect of model size is heterogeneous and model-dependent. Specially, scaling-up yields marginal performance gains for GloVe, Fast-

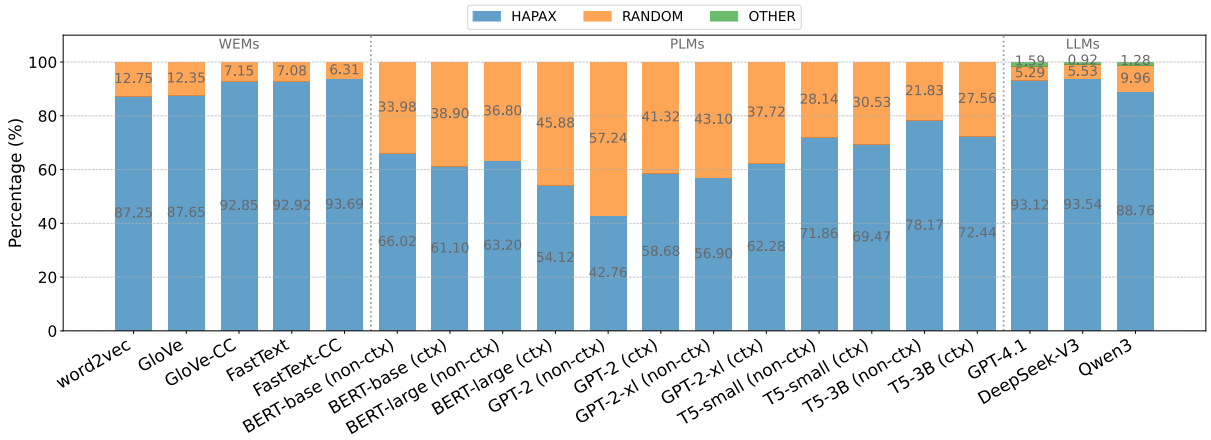


Figure 3: Error Percentages for Various Types of Wrong Hits in the Multiple-Choice Association Task. Blue bars show the percentage of HAPAX models deemed the most associated word with the stimulus, orange bars represent RANDOM hits, and green bars indicate other error types (e.g., multiple-word or out-of-choice generations).

Text, and GPT-2, but not for BERT or T5. This indicates that semantic-cognitive alignment relies more on architecture than on scale. It further suggests that parametric scaling laws interact differently with task-specific requirements.

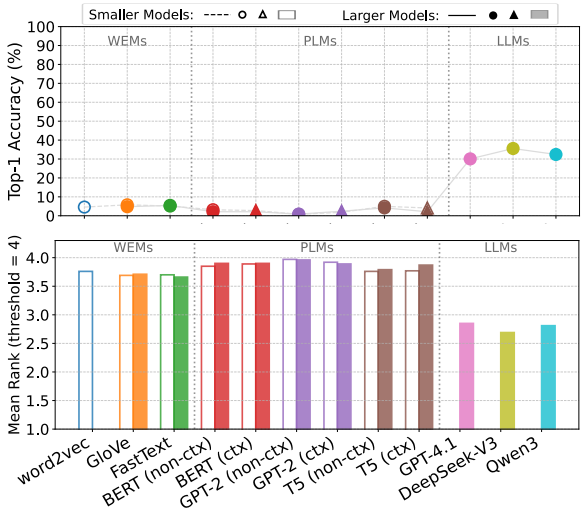


Figure 4: Top-1 Accuracies (above) and Mean Ranks (below) in the Open-Vocabulary Association Task.

4.3 Reverse Association

Task 3 requires two-step reasoning: first identifying the conceptual commonality among five related hint words, and then finding the target word connecting them from a broad candidate pool.

As shown in Figure 5 and 10, the results largely mirror the overall performance trends observed in Tasks 1 and 2, while further confirming that LLMs exhibit better alignment with human semantic knowledge. Specifically, LLMs achieve over 25% top-1 accuracies and demonstrate consistently lower mean ranks for the correct *Target* words as judged by humans. This suggests that LLMs are better equipped to handle tasks requiring abstract generalization and lexical retrieval.

Notably, static embeddings from WEMs also show relatively strong performance, achieving higher accuracy and lower average ranks compared to all PLMs. As for the vector representations from the latter type of models, it is possible that they are just too context-specific for tasks requiring to capture the semantics of word types.

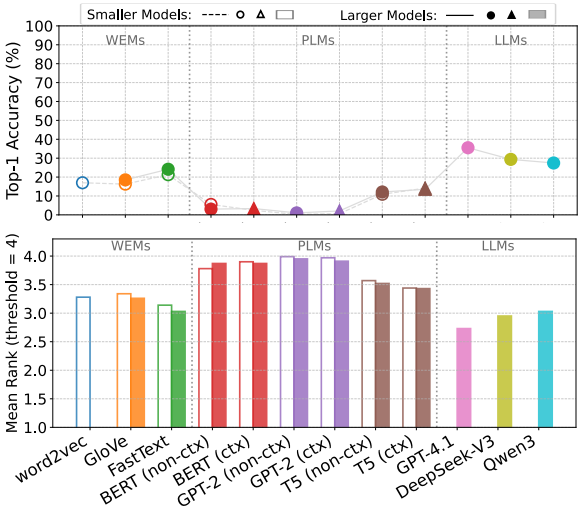


Figure 5: Top-1 Accuracies (above) and Mean Ranks (below) in the Reverse Association Task.

4.4 Concrete-Abstract Association

This task probes semantic space by comparing the strengths of inter-word semantic relationships within triads. As shown in Figure 6 and 11,

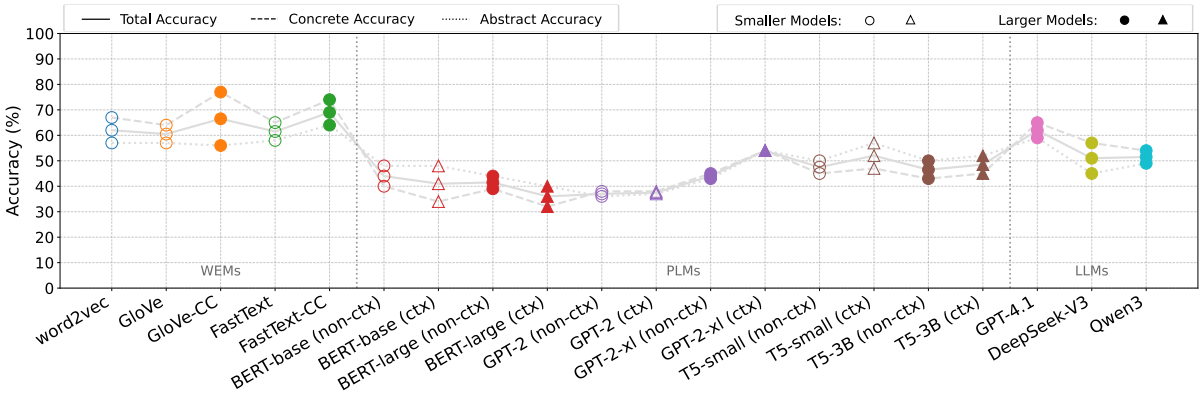


Figure 6: Accuracies in the Concrete-Abstract Association Task on Total, Concrete, and Abstract Datasets.

experimental results highlight the superior performance of WEMs, which significantly outperform embeddings from most PLMs, regardless of whether the word pairs are concrete or abstract. Moreover, regarding LLMs, the results also reveal that employing prompt-based methods on GPT-4.1 in this task achieves accuracy comparable to static embeddings derived from WEMs. In contrast, both DeepSeek-V3 and Qwen3 perform significantly worse—especially compared to larger WEMs, namely, GloVe-CC and FastText-CC, and their performance aligns more closely with that of T5 models among PLMs.

Interestingly, WEMs and LLMs show somewhat stronger performance on concrete triads than on abstract ones, while PLMs (like BERT and T5) exhibit the opposite pattern. This contrast may reflect their differing sensitivities to concreteness effects (Hill et al., 2014; Knupleš et al., 2023; Abramski et al., 2025), which describes that concrete words tend to evoke stronger but fewer associations, whereas abstract words elicit weaker but more diffuse associations. In this light, WEMs and LLMs are more effective at leveraging the focused, robust relationships typical of concrete concepts, whereas token-based embeddings from PLMs show fairly poor capability of adapting to such associations.

At last, we observe that incorporating contextual information during embedding extraction from PLMs leads to little performance degradation in BERT models but a slight improvement in GPT-2 and T5 models. However, these differences stemming from their distinct model architectures (Qiu et al., 2020) are not significant in this task. Besides, while scaling has minimal impact on PLMs' performance, it significantly enhances that of WEMs.

4.5 Remote Association

Contrary to expectations, the increased conceptual distances for the triads in Task 5, which may present greater challenges for human participants, have only a limited impact on the accuracies achieved by most language models when compared to the baseline results in Task 4. The results in Figure 12 indicate that significant accuracy differences arise only between WEMs and two types of PLMs (BERT and GPT-2), as well as two LLMs (DeepSeek-V3 and Qwen3). The top-performing models in each group remain consistent with those identified in other tasks, namely, FastText-CC among WEMs, T5-3B among PLMs, and GPT-4.1 among LLMs.

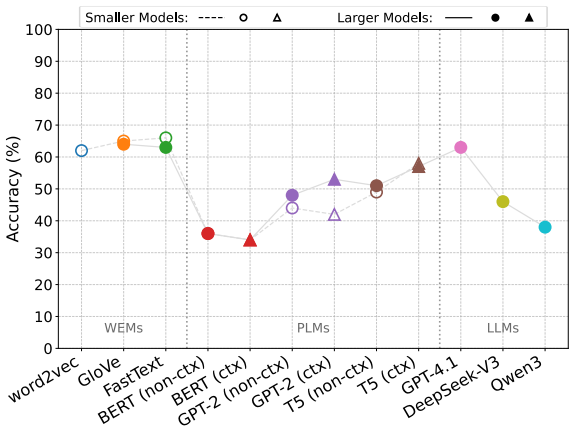


Figure 7: Accuracies in the Remote Association Task.

For this task, neither model size nor contextualization substantially affects the ability of WEMs and PLMs to identify intricate relational abstractions. Two primary factors may explain this finding. The first is the limited dataset size of 100 items, which may restrict generalization and robust statistical analysis. Second, theoretically, the remote

associations present in these triads generate scenarios that extend beyond textual information by incorporating not only perceptual but multimodal concerns, which may reduce the influence of differences in textual data. For instance, in the triad: A-*fear*, B-*guest*, C-*price* (P_{AB}=0.275, P_{AC}=0.425, P_{BC}=0.300), models with limited abstraction capabilities fail to identify the most human-like connection between fear and price, a subtle relationship that likely reflects real-world consumption and market experiences but is uncommon in training data. In such cases, evaluating atypicality by analyzing the distribution of model choices across both typical and atypical human responses, rather than relying solely on accuracy based on the most frequent human response, may yield a more informative comparison.

5 Conclusion

This study systematically evaluates the intrinsic semantic capabilities of diverse language models, including WEMs, PLMs, and LLMs, by leveraging their typical operational modes (e.g., word embeddings vs. prompt-based generation). Through the adaptation and integration of five kinds of classical psycholinguistic WATs, we assess how well these models perform on cognitively motivated benchmarks. The results reveal distinct performance and limitations across architectures and configurations.

First, WEMs and LLMs demonstrate better alignment with human association norms compared to PLMs, particularly in tasks requiring stable type-level semantic representations. Notably, LLMs outperform the other models in word retrieval (Tasks 1–2, similarity-dominant; Task 3, considering both similarity and relatedness), while WEMs do better in concept pairing (Tasks 4–5, relatedness-dominant), highlighting their complementary strengths across model architectures and the fact that human mental lexicon is good at synergizing similarity and relatedness, but not artificial systems. For WMEs, increasing model size generally improves performance. However, PLMs exhibit architecture-dependent behaviors in terms of scaling and contextualization: encoder-only models like BERT often degrade with larger scales and added contexts but decoder-only models (e.g., GPT-2) tend to benefit from both. For encoder–decoder models (e.g., T5), the impacts are task-specific. Their performance notably improves in Tasks 1 and 3 in these two settings but declines in Task 2.

LLMs' partial success in some WATs by mimicking human semantic behaviors demystifies the claim of their human-like intelligence. Yet they still struggle to fully replicate the versatility of the human mental lexicon, particularly in associating remote or abstract concepts. This suggests a tension between accuracy and creativity in language modeling, warranting deeper exploration. Together, these findings provide comprehensive insights into the alignment between language models and human cognition and highlight the value of psycholinguistic data for diagnosing model capabilities and biases.

Limitations

While this work provides broad insights into the semantic quality of different language models, it is limited by a few reasons for further improvement in the future.

A primary limitation of this study is the use of different evaluation methods across model types: cosine similarity for WEMs and PLMs, versus prompting for LLMs. While these approaches reflect typical usage patterns, the inconsistency challenges the validity of direct comparisons. Embedding similarity may capture relations beyond associative knowledge in some cases, whereas prompting can advantage LLMs by providing task-specific guidance. Consequently, some performance differences may reflect evaluation methods rather than intrinsic disparities in model knowledge. Future work should seek to standardize protocols, for example, by incorporating embedding-based measures for LLMs.

Additionally, while cosine distances are the most commonly used method for measuring semantic similarity between vectors, it has been criticized for potentially yielding arbitrary and meaningless "similarities" (Steck et al., 2024). Meanwhile, it may underestimate the actual similarity between contextualized embeddings (Wannasuphoprasit et al., 2023; Ijebu et al., 2025) and does not reliably indicate human associations due to its symmetric nature (Abramski et al., 2025). This limitation may impact our findings regarding the alignment between human assessments and the embeddings of WEMs and PLMs. Therefore, alternative methods, such as the soft cosine similarity proposed by Ijebu et al. (2025) or rank-based metrics (Santus et al., 2016, 2018; Zhelezniak et al., 2019), could be explored for a more robust investigation.

Also, our analysis of PLM models focused only on final-layer embeddings obtained through mean pooling, overlooking potential variations across transformer layers. Previous research suggested that intermediate layers may better capture lexical semantics (Ormerod et al., 2024). Additionally, it could be the case that our generic contexts were not informative enough to create robust representations, and better results might be achieved by sampling random sentence contexts with the target word from a large-scale corpus to represent and by averaging the corresponding embeddings (Bommasani et al., 2020; A et al., 2024; Nie et al., 2025). We will examine more layer-wise semantic properties and assess methods for distilling contextualized embeddings into static ones in the future. On the other hand, we also believe that this issue confirms that PLMs are probably not the best choice for the automatic collection of word associations, compared to WEMs and LLMs, given that researchers would have to perform the additional steps of context sampling and selection of the optimal layers.

Furthermore, the current study primarily focused on English WATs and did not adequately address advanced reasoning models and better configurations for prompting LLMs, which require further examination and comparison, including in multilingual and low-resource language contexts.

Finally, this study was conducted solely on semantic-level word associations. To gain a more in-depth understanding of language associations, future work can incorporate perspectives from other linguistic dimensions, such as morphological and phonological associations.

References

Pranav A, Yan Cong, Emmanuele Chersoni, Yu-Yin Hsu, and Alessandro Lenci. 2024. Comparing static and contextual distributional semantic models on intrinsic tasks: An evaluation on Mandarin Chinese datasets. In *Proceedings of LREC-COLING 2024*, pages 3610–3627, Torino, Italia. ELRA and ICCL.

Katherine Abramski, Riccardo Improta, Giulio Rossetti, and Massimo Stella. 2025. The "LLM World of Words" English Free Association Norms Generated by Large Language Models. *Scientific Data*, 12(1):1–9.

Fatemah Almeman and Luis Espinosa-Anke. 2024. GEAR: A Simple GENERATE, EMBED, AVERAGE AND RANK Approach for Unsupervised Reverse Dictionary. In *Proceedings of LREC-COLING*.

Marianna Apidianaki. 2023. From word types to tokens and back: A survey of approaches to word meaning representation and interpretation. *Computational Linguistics*, 49(2):465–523.

Xuechunzi Bai, Angelina Wang, Ilia Sucholutsky, and Thomas L Griffiths. 2025. Explicitly unbiased large language models still form biased associations. *Proceedings of the National Academy of Sciences*, 122(8).

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland. Association for Computational Linguistics.

Roger E Beaty and Yoed N Kenett. 2023. Associative thinking at the core of creativity. *Trends in cognitive sciences*, 27(7):671–683.

Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300.

Daniel Biś, Maksim Podkorytov, and Xiuwen Liu. 2021. Too much in common: Shifting of embeddings in transformer language models and its implications. In *Proceedings of NAACL 2021*, pages 5117–5130, Online. Association for Computational Linguistics.

Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758–4781, Online. Association for Computational Linguistics.

Jose Camacho-Collados, Claudio Delli Bovi, Luis Espinosa-Anke, Sergio Oramas, Tommaso Pasini, Enrico Santus, Vered Shwartz, Roberto Navigli, and Horacio Saggion. 2018. SemEval-2018 Task 9: Hypernym Discovery. In *Proceedings of SemEval*.

Tanguy Cazalets and Joni Dambre. 2025. Word synchronization challenge: A benchmark for word association responses for large language models. In *International Conference on Human-Computer Interaction*, pages 3–19. Springer.

Kenneth Church and Patrick Hanks. 1990. Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics*, 16(1):22–29.

Salvatore Citraro, Michael S Vitevitch, Massimo Stella, and Giulio Rossetti. 2023. Feature-rich multiplex lexical networks reveal mental strategies of early language learning. *Scientific Reports*, 13(1):1474.

Stephen Clark. 2015. Vector space models of lexical meaning. *The Handbook of Contemporary semantic theory*, pages 493–522.

Simon De Deyne, Chunhua Liu, and Lea Frermann. 2024. Can GPT-4 Recover Latent Semantic Relational Information from Word Associations? A Detailed Analysis of Agreement with Human-annotated Semantic Ontologies. In *Proceedings of the COLING Workshop on Cognitive Aspects of the Lexicon*.

Simon De Deyne, Danielle J Navarro, Guillem Collell, and Andrew Perfors. 2021. Visual and Affective Multimodal Models of Word Meaning in Language and Mind. *Cognitive Science*, 45(1):e12922.

Simon De Deyne, Danielle J Navarro, Amy Perfors, Marc Brysbaert, and Gert Storms. 2019. The "small world of words" english word association norms for over 12,000 cue words. *Behavior research methods*, 51:987–1006.

Simon De Deyne, Amy Perfors, and Daniel J Navarro. 2016. Predicting Human Similarity Judgments with Distributional Models: The Value of Word Associations. In *Proceedings of COLING 2016*.

DeepSeek-AI. 2024. Deepseek-v3 technical report.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL 2019*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of EMNLP-IJCNLP*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. 2019. Understanding undesirable word embedding associations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1696–1705, Florence, Italy. Association for Computational Linguistics.

Stefan Evert and Gabriella Lapesa. 2021. FAST: A carefully sampled and cognitively motivated dataset for distributional semantic evaluation. In *Proceedings of CONLL*.

Zhaoxin Feng, Jianfei Ma, Emmanuele Chersoni, Xiaojing Zhao, and Xiaoyi Bao. 2025. Learning to Look at the Other Side: A Semantic Probing Study of Word Embeddings in LLMs with Enabled Bidirectional Attention. In *Proceedings of ACL*.

John Rupert Firth. 1957. *A Synopsis of Linguistic Theory 1930–55*. Longmans.

James Fodor, Simon De Deyne, and Shinsuke Suzuki. 2023. The importance of context in the evaluation of word embeddings: The effects of antonymy and polysemy. In *Proceedings of IWCS*, pages 155–172, Nancy, France. Association for Computational Linguistics.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15.

Thilo Hagendorff. 2023. Machine Psychology: Investigating Emergent Capabilities and Behavior in Large Language Models Using Psychological Methods. *arXiv preprint arXiv:2303.13988*, 1.

Felix Hill, Anna Korhonen, and Christian Bentz. 2014. A quantitative empirical analysis of the abstract/concrete distinction. *Cognitive science*, 38(1):162–177.

Funebi Francis Ijebu, Yuanchao Liu, Chengjie Sun, and Patience Usoro Usip. 2025. Soft cosine and extended cosine adaptation for pre-trained language model semantic vector analysis. *Applied Soft Computing*, 169:112551.

Dan R Johnson and Richard W Hass. 2022. Semantic context search in creative idea generation. *The Journal of Creative Behavior*, 56(3):362–381.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Yoed N. Kenett and Miriam Faust. 2019. A semantic network cartography of the creative mind. *Trends in Cognitive Sciences*, 23(4):271–274.

George R Kiss, Christine Armstrong, Robert Milroy, and James Piper. 1973. An associative thesaurus of english and its computer analysis. *The Computer and Literary Studies*, 153.

Urban Knupleš, Diego Frassinelli, and Sabine Schulte im Walde. 2023. Investigating the Nature of Disagreements on Mid-Scale Ratings: A Case Study on the Abstractness-Concreteness Continuum. In *Proceedings of CONLL*.

Trina Kwong, Emmanuele Chersoni, and Rong Xiang. 2022. Evaluating monolingual and crosslingual embeddings on datasets of word association norms. In *Proceedings of the BUCC Workshop within LREC 2022*, pages 1–7, Marseille, France. European Language Resources Association.

Alessandro Lenci, Magnus Sahlgren, Patrick Jeuniaux, Amaru Cuba Gyllensten, and Martina Miliani. 2022. A comparative evaluation and analysis of three generations of distributional semantic models. *Language resources and evaluation*, 56(4):1269–1313.

Lorenz Linhardt, Tom Neuhäuser, Lenka Tětková, and Oliver Eberle. 2025. Cat, Rat, Meow: On the Alignment of Language Model and Human Term-Similarity Judgments. In *Proceedings of the ICLR Workshop on Re-Align*.

Lea Löhn, Niklas Kiehne, Alexander Ljapunov, and Wolf-Tilo Balke. 2024. Is Machine Psychology here? On Requirements for Using Human Psychological Tests on Large Language Models. In *Proceedings of INLG*.

Hongjing Lu, Nicholas Ichien, and Keith J Holyoak. 2022. Probabilistic analogical mapping with semantic relation networks. *Psychological review*, 129(5):1078.

Paweł Mandera, Emmanuel Keuleers, and Marc Brysbaert. 2017. Explaining Human Performance in Psycholinguistic Tasks with Models of Semantic Similarity Based on Prediction and Counting: A Review and Empirical Validation. *Journal of Memory and Language*, 92:57–78.

Youness Mansar, Juyeon Kang, and Ismail El Maarouf. 2021. The FinSim-2 2021 Shared Task: Learning semantic similarities for the financial domain. In *Companion Proceedings of the Web Conference 2021*, pages 288–292.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Ken McRae, Saman Khalkhali, and Mary Hare. 2012. Semantic and Associative Relations in Adolescents and Young Adults: Examining a Tenuous Dichotomy. *Psychology Publications*, 115.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Workshop Track Proceedings*.

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

Douglas L Nelson, Cathy L McEvoy, and Thomas A Schreiber. 2004. The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, 36(3):402–407.

Zhijie Nie, Zhangchi Feng, Mingxin Li, Cunwang Zhang, Yanzhao Zhang, Dingkun Long, and Richong Zhang. 2025. When text embedding meets large language model: A comprehensive survey.

Mark Ormerod, Jesús Martínez del Rincón, and Barry Devereux. 2024. How is a "kitchen chair" like a "farm horse"? exploring the representation of noun-noun compound semantics in transformer-based language models. *Computational Linguistics*, 50(1):49–81.

Max Peeperkorn, Tom Kouwenhoven, Dan Brown, and Anna Jordanous. 2024. Is temperature the creativity parameter of large language models? In *Proceedings of ICCC'24*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of EMNLP 2014*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China technological sciences*, 63(10):1872–1897.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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.

Reinhard Rapp and Michael Zock. 2014. The CogALex-IV shared task on the lexical access problem. In *Proceedings of the 4th Workshop on Cognitive Aspects of the Lexicon (CogALex)*, pages 1–14, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Prisha Samdarshi, Mariam Mustafa, Anushka Kulkarni, Raven Rothkopf, Tuhin Chakrabarty, and Smaranda Muresan. 2024. Connecting the dots: Evaluating abstract reasoning capabilities of llms using the new york times connections word game. In *Proceedings of EMNLP 2024*, pages 21219–21236.

Enrico Santus, Emmanuele Chersoni, Alessandro Lenci, Chu-Ren Huang, and Philippe Blache. 2016. Testing apsyn against vector cosine on similarity estimation. In *Proceedings of PACLIC*.

Enrico Santus, Hongmin Wang, Emmanuele Chersoni, and Yue Zhang. 2018. A rank-based similarity metric for word embeddings. In *Proceedings of ACL*.

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*, pages 887–890.

William Timkey and Marten van Schijndel. 2021. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In *Proceedings of EMNLP*, pages 4527–4546, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xueyang Wang, Qunlin Chen, Kaixiang Zhuang, Jingyi Zhang, Robert A Cortes, Daniel D Holzman, Li Fan, Cheng Liu, Jiangzhou Sun, Xianrui Li, et al. 2024. Semantic associative abilities and executive control functions predict novelty and appropriateness of idea generation. *Communications Biology*, 7(1):703.

Saeth Wannasuphoprasit, Yi Zhou, and Danushka Bollegala. 2023. Solving cosine similarity underestimation between high frequency words by ℓ_2 norm discounting. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8644–8652, Toronto, Canada. Association for Computational Linguistics.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

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, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report.

Wenjing Yang, Adam E. Green, Qunlin Chen, Yoed N. Kenett, Jiangzhou Sun, Dongtao Wei, and Jiang Qiu. 2022. Creative problem solving in knowledge-rich contexts. *Trends in Cognitive Sciences*, 26(10):849–859.

Zhong Zhang, Chongming Gao, Cong Xu, Rui Miao, Qinli Yang, and Junming Shao. 2020. Revisiting representation degeneration problem in language modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 518–527, Online. Association for Computational Linguistics.

Vitalii Zhelezniak, Aleksandar Savkov, April Shen, and Nils Y Hammerla. 2019. Correlation coefficients and semantic textual similarity. In *Proceedings of NAACL*.

Michael Zock. 2002. Sorry, What Was Your Name Again, or How to Overcome the Tip-of-the tongue Problem with the Help of a Computer? In *Proceedings of the SEMANET Workshop on Building and Using Semantic Networks*.

A Discussion on the Properties of Different WATs

Studies of semantic knowledge in vector spaces typically use two key metrics: *semantic similarity* and *semantic relatedness* (Fodor et al., 2023). The former means the degree of interchangeability between words based on their core meanings (Miller and Charles, 1991), as exemplified by *accept* and *receive* due to their overlapping meanings. In contrast, the latter encompasses broader conceptual connections, including functional, contextual, or psychological associations, even when words exhibit minimal semantic overlap (Gladkova et al., 2016). For instance, *air* and *plane* demonstrate high relatedness despite low similarity. These dimensions are rooted in lexical networks together, with different word association tasks highlighting distinct aspects.

Tasks 1 and 2 primarily assess semantic similarity, as they require models to identify the most semantically proximate word to a given stimulus. In Task 1, the *FIRST* response exhibits high interchangeability with the given stimulus (e.g., *receive* for *accept*), conforming to Miller and Charles (1991)'s definition of semantic similarity. While Task 2 employs an open-vocabulary paradigm, it requires the generation or selection of maximally similar words, maintaining its focus on direct meaning alignment. Both tasks prioritize paradigmatic relations (synonymy or near-synonymy).

Tasks 4 and 5 are relatedness-focused ones due to their emphasis on detecting implicit conceptual connections beyond semantic interchangeability. The triad tasks (Concrete-Abstract and Remote) evaluate models' ability to identify word pairs based on latent relational features. For instance, [*banana*, *cherry*, *pineapple*] in Task 4 (*banana* and *pineapple* are regarded as the most related concepts, but they have totally different denotations), and [*fence*, *mask*, *salt*] in Task 5 (the first two words are most related but non-interchangeable).

Different from the aforementioned tasks, Task 3 requires models to simultaneously make judgments on semantic similarity and relatedness, as illustrated through examples from the CogALex dataset (Rapp and Zock, 2014). The case of [*plenty*, *many*, *lots*, *around*, *leap* $\rightarrow$ *abound*] demonstrates similarity-driven processing, where identifying the *Target* depends on recognizing shared core meanings of quantitative abundance. In contrast, the example [*plane*, *fresh*, *water*, *breathe*, *force* $\rightarrow$ *air*]

reveals their internal relatedness through its web of diverse associations, including functional, ecological, physical, and perceptual connections.

Our findings echo prior work (Lenci et al., 2022; A et al., 2024) on the semantic representation capabilities of WEMs versus contextualized models (PLMs/LLMs). We found that the distinction between association tasks via semantic similarity and relatedness is highly significant as it offers a clearer framework for comparing architectures, emphasizing that human cognition seamlessly combines similarity and relatedness, while language models lag behind and show different limitations.

B Results of Multiple-Choice Association

Table 2: Accuracies (Acc.) and Frequencies of Incorrect Responses (*HAPAX*, *RANDOM*, and *OTHER*) in Task 1.

Types	Settings	Models	Acc. (%)	*HAPAX*	*RANDOM*	*OTHER*
WMEs	embeddings	word2vec	77.90	2,203	322	
		GloVe	79.31	2,072	292	
		GloVe-CC	80.28	2,092	161	
		FastText	82.07	1,904	145	
		FastText-CC	83.34	1,783	120	
PLMs	non-contextualized embeddings	BERT-base	58.26	3,150	1,621	
		BERT-large	52.81	3,409	1,985	
		GPT-2	34.23	3,215	4,303	
		GPT-2-xl	52.42	3,094	2,344	
		T5-small	65.89	2,801	1,097	
		T5-3B	67.05	2,944	822	
	contextualized embeddings	BERT-base	52.01	3,352	2,134	
		BERT-large	45.46	3,374	2,860	
		GPT-2	52.25	3,203	2,255	
		GPT-2-xl	62.55	2,666	1,615	
		T5-small	67.47	2,583	1,135	
		T5-3B	71.34	2,373	903	
LLMs	prompt	GPT-4.1	86.77	1,408	80	24
		DeepSeek-V3	86.72	1,420	84	14
		Qwen3	79.53	2,077	233	30

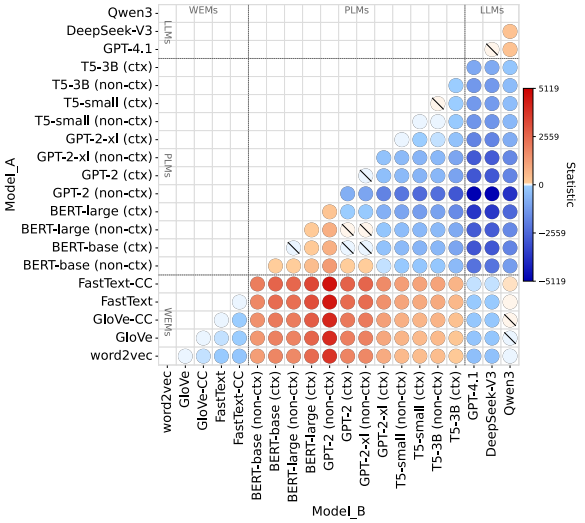


Figure 8: Pairwise McNemar's Tests on Task 1 ($p < 0.05$). Colored cells denote the significantly stronger models based on accuracies: red for Model_A and blue for Model_B. Dashes indicate non-significant differences.

C Results of Open-Vocabulary Association

Table 3: Top-1 Accuracies (Top-1 Acc.) and Mean Ranks with the Threshold of 4 (MR/4) in Task 2.

Types	Settings	Models	Top-1 Acc. (%)	MR/4
WMEs	embeddings	word2vec	4.59	3.76
		GloVe	5.78	3.69
		GloVe-CC	4.79	3.71
		FastText	5.14	3.70
		FastText-CC	5.49	3.66
PLMs	non-contextualized embeddings	BERT-base	3.19	3.85
		BERT-large	2.13	3.90
		GPT-2	0.78	3.97
		GPT-2-xl	0.86	3.96
		T5-small	4.99	3.76
		T5-3B	4.17	3.79
	contextualized embeddings	BERT-base	2.74	3.89
		BERT-large	2.13	3.90
		GPT-2	1.84	3.92
		GPT-2-xl	2.41	3.89
		T5-small	4.11	3.77
		T5-3B	2.12	3.87
LLMs	prompt	GPT-4.1	30.07	2.85
		DeepSeek-V3	35.56	2.69
		Qwen3	32.40	2.81

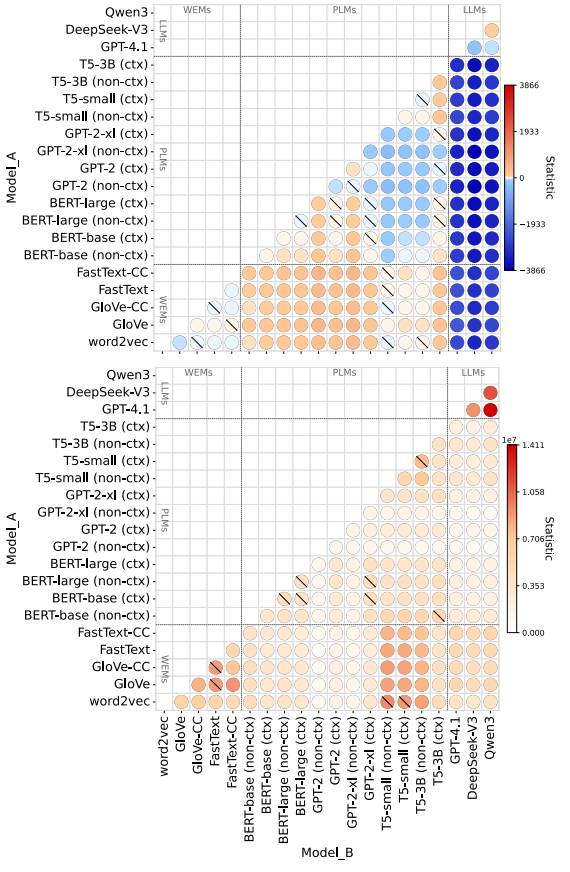


Figure 9: Pairwise McNemar's Tests (above) and Wilcoxon Signed-Rank Tests (below) on Task 2 ($p < 0.05$). For the plot above, colored cells denote the significantly stronger models based on top-1 accuracies: red for Model_A and blue for Model_B. For the below one, colored cells denote significant differences on *FIRST* ranks. Dashes indicate non-significant differences.

D Results of Reverse Association

Table 4: Top-1 Accuracies (Top-1 Acc.) and Mean Ranks with the Threshold of 4 (MR/4) in Task 3.

Types	Settings	Models	Top-1 Acc. (%)	MR/4
WMEs	embeddings	word2vec	16.99	3.28
		GloVe	16.27	3.34
		GloVe-CC	18.52	3.26
		FastText	21.26	3.14
		FastText-CC	24.14	3.03
PLMs	non-contextualized embeddings	BERT-base	5.53	3.78
		BERT-large	3.04	3.87
		GPT-2	0.25	3.99
		GPT-2-xl	1.04	3.95
		T5-small	10.90	3.57
		T5-3B	12.11	3.52
	contextualized embeddings	BERT-base	2.38	3.90
		BERT-large	3.34	3.87
		GPT-2	0.63	3.97
		GPT-2-xl	2.08	3.91
		T5-small	14.14	3.44
		T5-3B	13.51	3.43
LLMs	prompt	GPT-4.1	35.53	2.73
		DeepSeek-V3	29.37	2.95
		Qwen3	27.45	3.03

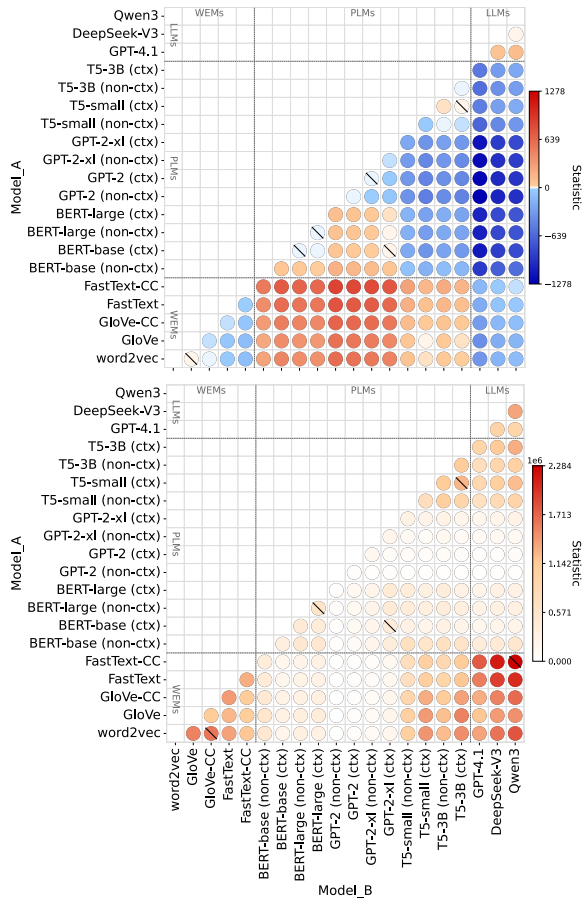


Figure 10: Pairwise McNemar's Tests (above) and Wilcoxon Signed-Rank Tests (below) on Task 3 ($p < 0.05$). For the plot above, colored cells denote the significantly stronger models based on top-1 accuracies: red for Model_A and blue for Model_B. For the below one, colored cells denote significant differences on *Target* ranks. Dashes indicate non-significant differences.

E Results of Concrete-Abstract Association

Table 5: Accuracies (Acc.) on Total (T), Concrete (C), and Abstract (A) datasets in Task 4.

Types	Settings	Models	T-Acc. (%)	C-Acc. (%)	A-Acc. (%)
WMEs	embeddings	word2vec	62.00	67.00	57.00
		GloVe	60.50	64.00	57.00
		GloVe-CC	66.50	77.00	56.00
		FastText	61.50	65.00	58.00
		FastText-CC	69.00	74.00	64.00
PLMs	non-contextualized embeddings	BERT-base	44.00	40.00	48.00
		BERT-large	41.50	39.00	44.00
		GPT-2	37.00	38.00	36.00
		GPT-2-xl	44.00	45.00	43.00
		T5-small	47.50	45.00	50.00
		T5-3B	46.50	43.00	50.00
	contextualized embeddings	BERT-base	41.00	34.00	48.00
		BERT-large	36.00	32.00	40.00
		GPT-2	37.50	38.00	37.00
		GPT-2-xl	54.00	54.00	54.00
		T5-small	52.00	47.00	57.00
		T5-3B	48.50	45.00	52.00
LLMs	prompt	GPT-4.1	62.00	65.00	59.00
		DeepSeek-V3	51.00	57.00	45.00
		Qwen3	51.50	54.00	49.00

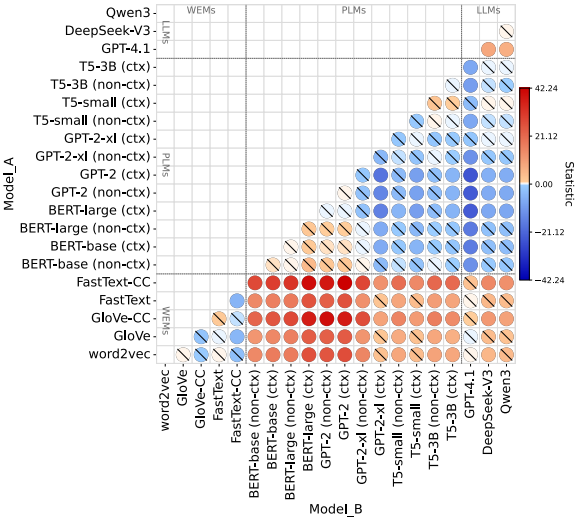


Figure 11: Pairwise McNemar's Tests on Task 4 ($p < 0.05$). Colored cells denote the significantly stronger models based on t-accuracies: red for Model_A and blue for Model_B. Dashes indicate non-significant differences.

F Results of Remote Association

G Ablation Studies on Prompting LLMs

We conducted exploratory experiments to examine how external (prompt design) and internal factors (temperature settings, reasoning modes) influence LLM performance across different WATs. Datasets applied here were randomly sampled from our main evaluation data as introduced in Table 1, with 200 items per task for Tasks 1-3, and full sets for Task 4 (200 items) and Task 5 (100 items).

Table 6: Accuracies (Acc.) in Task 5.

Types	Settings	Models	Acc. (%)
WMEs	embeddings	word2vec	62.00
		GloVe	65.00
		GloVe-CC	64.00
		FastText	66.00
		FastText-CC	63.00
PLMs	non-contextualized embeddings	BERT-base	36.00
		BERT-large	36.00
		GPT-2	44.00
		GPT-2-xl	48.00
		T5-small	49.00
		T5-3B	51.00
	contextualized embeddings	BERT-base	34.00
		BERT-large	34.00
		GPT-2	42.00
		GPT-2-xl	53.00
		T5-small	58.00
		T5-3B	57.00
LLMs	prompt	GPT-4.1	63.00
		DeepSeek-V3	46.00
		Qwen3	38.00

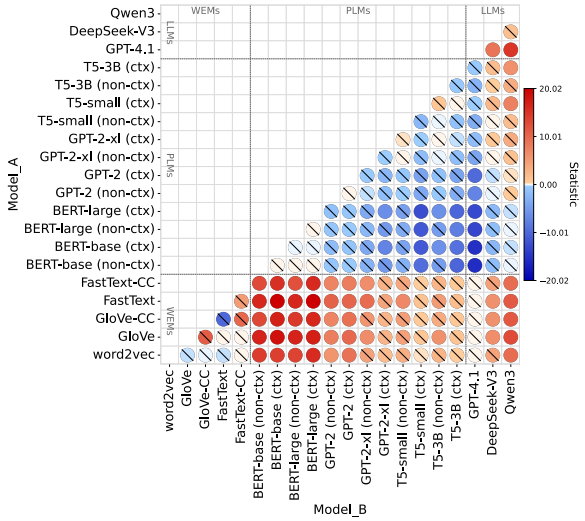


Figure 12: Pairwise McNemar's Tests on Task 5 ($p < 0.05$). Colored cells denote the significantly stronger models based on t-accuracies: red for Model_A and blue for Model_B. Dashes indicate non-significant differences.

G.1 Different Prompts: Zero-shot vs. Few-shot

Two sets of prompt instructions were designed by referring to those in the study of De Deyne et al. (2024), namely, 1) simple zero-shot prompts and 2) enhanced few-shot ones, detailed in Figures 13 to 18. The exemplars for few-shot prompts were sourced from established association norms such as EAT (Kiss et al., 1973), USF (Nelson et al., 2004), and SWOW (De Deyne et al., 2019), excluding any items overlapping with our evaluation datasets to prevent contamination. The temperature for this subexperiment was fixed at 0.01 and the reasoning mechanism was disabled to isolate prompt efficacy.

Results in Table 7 exhibit that detailed few-shot prompts consistently enhance LLM performance except in Task 2. For instance, GPT-4.1 achieves over 5% accuracy gains in Tasks 3 and 4, and DeepSeek-V3 and Qwen3 show even more than 10% improvements. However, the benefits of detailed few-shot prompting are model- and task-dependent, as evidenced by GPT-4.1's performance in Task 2, where such prompts had marginal or even negative effects.

G.2 Different Temperatures: 0.01 vs. 0.5 vs. 1

The temperature is a built-in parameter of LLMs to control the randomness and the so-called creativity of their outputs (Peeperkorn et al., 2024). It spans [0, 2] with higher values corresponding to increased diversity, while lower values yield more focused and deterministic outputs. It is assumed to have effects on models' semantic association capabilities, potentially mapping cognitive factors in human associative behavior. Therefore, we conducted subexperiment on comparing three temperatures: 0.01, 0.5, and 1 with simple zero-shot prompts and without the thinking mode.

Although the current test was limited to half of the full temperature range, Table 8 demonstrates two key observations: 1) Temperature effects vary across models and tasks, such as, GPT-4.1 achieves optimal performance at 0.01 and 0.5, DeepSeek-V3 benefits most from 0.5, Qwen3 performs better at 0.5 and 1, and Tasks 2 and 3 show robustness to 0.5 compared to other tasks; 2) Performance differences induced by different temperatures remain subtle (less than 5%) across all assessed models and tasks.

G.3 Different Modes: Standard vs. Reasoning

To investigate potential advantages of reasoning mechanisms, we conducted a subexperiment on Qwen3 with reasoning activation as the only variable, using zero-shot prompts and a fixed temperature of 0.01. Surprisingly, the reasoning is not advantageous in all WATs. Notably, in Tasks 2 and 4—abstract word pairing, enabling reasoning may lead to overthinking and hence misjudgments in semantic similarity and relatedness assessments.

Together above results unveil the versatility of the human associative ability, which cannot be fully reproduced by LLM configurations.

Table 7: Comparisons of LLM Results across Different Prompt Strategies. Boldface values indicate the highest performance achieved by the model on a given task across all strategies.

Tasks	Metrics	GPT-4.1		DeepSeek-V3		Qwen3	
		zero-shot	few-shot	zero-shot	few-shot	zero-shot	few-shot
Task 1	Acc. (%)	90.00	**91.00**	89.50	**90.50**	84.50	**89.50**
Task 2	Top-1 Acc. (%)	**31.00**	30.50	35.50	**36.50**	32.50	**35.00**
	MR/4	**2.75**	2.76	2.68	**2.53**	2.78	**2.61**
Task 3	Top-1 Acc. (%)	32.50	**37.00**	26.00	**37.50**	23.00	**34.00**
	MR/4	2.86	**2.72**	3.03	**2.71**	3.19	**2.82**
Task 4	T-Acc. (%)	62.00	**67.50**	50.50	**75.00**	51.50	**61.50**
	C-Acc. (%)	65.00	**66.00**	56.00	**80.00**	54.00	**58.00**
	A-Acc. (%)	59.00	**69.00**	45.00	**70.00**	49.00	**65.00**
Task 5	Acc. (%)	63.00	**68.00**	46.00	**60.00**	38.00	**43.00**

Table 8: Comparisons of LLM Results across Different Temperature Settings. Boldface values indicate the highest performance achieved by the model on a given task across all settings.

Tasks	Metrics	GPT-4.1			DeepSeek-V3			Qwen3		
		0.01	0.5	1	0.01	0.5	1	0.01	0.5	1
Task 1	Acc. (%)	**90.00**	89.50	88.50	**89.50**	88.50	88.50	84.50	**86.50**	86.00
Task 2	Top-1 Acc. (%)	31.00	**32.00**	30.00	35.50	**37.00**	34.50	32.50	**35.00**	32.00
	MR/4	2.80	**2.78**	2.81	2.68	**2.60**	2.66	2.78	**2.71**	**2.71**
Task 3	Top-1 Acc. (%)	32.50	**36.00**	33.00	26.00	**31.50**	29.00	23.00	**26.50**	**26.50**
	MR/4	2.83	**2.76**	2.80	3.03	**2.93**	2.98	3.19	3.11	**3.06**
Task 4	T-Acc. (%)	**62.00**	57.00	59.00	51.00	**51.50**	51.00	51.50	52.50	**52.50**
	C-Acc. (%)	**65.00**	58.00	63.00	**57.00**	56.00	**57.00**	54.00	**55.00**	52.00
	A-Acc. (%)	**59.00**	56.00	55.00	45.00	**47.00**	45.00	49.00	50.00	**53.00**
Task 5	Acc. (%)	63.00	**67.00**	63.00	**46.00**	43.00	45.00	38.00	**39.00**	35.00

Table 9: Comparisons of Qwen3 Results with Different Thinking Modes. Boldface values indicate the highest performance achieved by the model on a given task within two modes.

Tasks	Metrics	Qwen3	
		standard	reasoning
Task 1	Acc. (%)	84.50	**89.00**
Task 2	Top-1 Acc. (%)	**32.50**	28.50
	MR/4	**2.78**	2.84
Task 3	Top-1 Acc. (%)	23.00	**28.00**
	MR/4	3.19	**3.01**
Task 4	T-Acc. (%)	51.50	**52.00**
	C-Acc. (%)	54.00	**57.00**
	A-Acc. (%)	**49.00**	47.00
Task 5	Acc. (%)	38.00	**45.00**

***** Simple Zero-Shot Prompt *****

System: You are a native speaker of English participating in a psycholinguistic test about word meaning.

User:

**** Task 1 ****
- You will be presented with a list of words separated by "-" that consists of a cue (the first one) and three candidates.
- You are asked to choose one target candidate from the three given candidates that is most closely associated with the cue.
- Remember to only respond with one target candidate word and do not further elaborate on your response.
- Format your response as json: {cue-candidate1-candidate2-candidate3: target candidate}.

- Input:{input}
- Output:

**** Task 2 ****
- You will be presented with a cue word.
- You are asked to output a list consisting of thirty words that are most closely associated with the cue word.
- Rank all thirty words according to their strength of association with the cue words in descending order.
- Remember to only respond with one list of ranked words and do not further elaborate on your response.
- Format your response as json: {cue: [response1, response2, ..., response30]}.

- Input:{input}
- Output:

**** Task 3 ****
- You will be presented with five hint words seperated by "-".
- You are asked to output a list consisting of thirty words that are most closely associated with the given five hint words
- Rank all thirty words according to their strength of association with all five hint words in descending order.
- Remember to only respond with one list of ranked words and do not further elaborate on your response.
- Format your response as json: {word1-word2-word3-word4-word5: [response1, response2, ..., response30]}.

- Input:{input}
- Output:

**** Task 4 ****
- You will be presented with a triplet of words that can be marked as "A", "B", "C" in sequence.
- You are asked to output a list consisting of three alphabetic pairs that are ranked with the strength of word association within their corresponding word pairs.
- Remember to only respond with one list of ranked pairs and do not further elaborate on your response.
- Format your response as json: {wordA-wordB-wordC:["AB", "BC", "AC"]}.

- Input:{input}
- Output:

**** Task 5 ****
- You will be presented with a triplet of words that can be marked as "A", "B", "C" in sequence.
- You are asked to output a list consisting of three alphabetic pairs that are ranked with the strength of word association within their corresponding word pairs.
- Remember to only respond with one list of ranked pairs and do not further elaborate on your response.
- Format your response as json: {wordA-wordB-wordC:["AB", "BC", "AC"]}.

- Input:{input}
- Output:

Figure 13: Simple Zero-shot Prompt Instructions for LLMs across Five WATs.

***** Enhanced Few-Shot Prompt – Task 1 *****

System: You are functioning as a native English speaker with unimpaired lexical access capabilities participating in a controlled psycholinguistic experiment. Your task requires making semantic association judgments through systematic cognitive operations.

User:

- This test is called "Multiple-Choice Word Association", designed to measure your ability to associate words with each other from a restricted list.
- You will be presented with a list of words separated by "-" that consists of a cue (priming lexical item) in the first position and three candidates (a triplet of potential association targets) in the second to fourth positions.
- You are asked to choose one target candidate from the three given candidates that is most closely associated with the cue in consideration of semantic (denotative overlap), conceptual (connotative alignment) and cognitive (co-occurrence frequency) association strengths.
- Remember to only respond with one target candidate word and do not further elaborate on your response.
- Format your response as json: {cue-candidate1-candidate2-candidate3: target candidate}.

- Here are some examples:
- {
- "input": "fibre-moral-glass-cries",
- "output": {"fibre-moral-glass-cries": "glass"}
- },
- {
- "input": "alert-jagger-inactive-awake",
- "output": {"alert-jagger-inactive-awake": "awake"}
- },
- {
- "input": "poison-arsenic-milford-shakespeare",
- "output": {"poison-arsenic-milford-shakespeare": "arsenic"}
- }

- Input:{input}
- Output:

Figure 14: Extended Few-shot Prompt Instructions for LLMs in Task 1: Multiple-Choice Association.

***** Enhanced Few-Shot Prompt – Task 2 *****

System: You are functioning as a native English speaker with unimpaired lexical access capabilities participating in a controlled psycholinguistic experiment. Your task requires making semantic association judgments through systematic cognitive operations.

User:

- This test is called "Open-Vocabulary Word Association", designed to measure your ability to perform deep semantic network traversal.
- You will be presented with a cue word.
- You are asked to output a list consisting of thirty words that are most closely associated with the cue word in consideration of semantic (denotative overlap), conceptual (connotative alignment) and cognitive (co-occurrence frequency) association strengths.
- Rank all thirty words according to their strength of association with the cue words in descending order.
- Remember to only respond with one list of ranked words and do not further elaborate on your response.
- Format your response as json: {cue: [response1, response2, ..., response30]}.

- Here are some examples:
- {
- "input": "fibre",
- "output": {{"fibre":["food", "cloth", "cereal", "fabric", "optic", "diet", "cotton", "glass", "poop", "internet", "bread", "bran", "optics", "material", "hair", "thread", "health", "strength", "rope", "wheat", "clothes", "grain", "wool", "clothing", "textile", "wire", "healthy", "paper", "digestion", "laxative"]}
- },
- {
- "input": "alert",
- "output": {{"alert":["awake", "alarm", "red", "aware", "fire", "siren", "warning", "ready", "warn", "danger", "attention", "attentive", "coffee", "light", "notice", "conscious", "morning", "observant", "sharp", "tense", "lights", "know", "keen", "emergency", "high", "caution", "mind", "tell", "reminder", "vigilant"]}
- },
- {
- "input": "poison",
- "output": {{"poison":["death", "Ivy", "kill", "apple", "arsenic", "liquid", "bottle", "bad", "snake", "drink", "venom", "deadly", "green", "rat", "dart", "dangerous", "chemical", "frog", "danger", "sickness", "mushroom", "murder", "toxic", "food", "fish", "band", "die", "rats", "evil", "crossbones"]}
- }

- Input:{input}
- Output:

Figure 15: Extended Few-shot Prompt Instructions for LLMs in Task 2: Open-Vocabulary Association.

***** Enhanced Few-Shot Prompt – Task 3 *****

System: You are functioning as a native English speaker with unimpaired lexical access capabilities participating in a controlled psycholinguistic experiment. Your task requires making semantic association judgments through systematic cognitive operations.

User:

- This test is called "Reverse Word Association", designed to measure your ability to address the word access problem by predicting the trigger based on the commonality between given words.
- You will be presented with five hint words separated by "-".
- You are asked to output a list consisting of thirty words that are most closely associated with the given five hint words in consideration of semantic (denotative overlap), conceptual (connotative alignment) and cognitive (co-occurrence frequency) association strengths.
- Rank all thirty words according to their strength of association with all five hint words in descending order.
- Remember to only respond with one list of ranked words and do not further elaborate on your response.
- Format your response as json: {word1-word2-word3-word4-word5: [response1, response2, ..., response30]}.

- Here are some examples:
- {
- "input": "together-joined-effort-harvester-honours",
- "output": {{"together-joined-effort-harvester-honours":["combined", "mixed", "mix", "added", "two", "bound", "sum", "multiple", "joint", "total", "linked", "stuck", "join", "harvester", "pair", "words", "with", "connected", "baking", "score", "paired", "grouped", "eggs", "combine", "associated", "amalgamation", "amalgamated", "one", "attached", "integration"]}}
- },
- {
- "input": "centre-end-earth-East-man",
- "output": {{"centre-end-earth-East-man":["middle", "child", "average", "central", "between", "median", "name", "age", "school", "class", "finger", "top", "last", "bottom", "waist", "road", "medium", "ages", "half", "ground", "compromise", "start", "stuck", "sister", "surrounded", "sandwich", "muddle", "first", "amid", "inside"]}}
- },
- {
- "input": "to-should-not-must-nought",
- "output": {{"to-should-not-must-nought":["ought", "zero", "need", "will", "would", "obligation", "might", "guilt", "obligated", "eight", "right", "could", "require", "shall", "thought", "responsibility", "proper", "old", "fashioned", "nothing", "can", "caught", "grandfather", "go", "duty", "supposed"]}}
- }

- Input:{input}
- Output:

Figure 16: Extended Few-shot Prompt Instructions for LLMs in Task 3: Reverse Association.

***** Enhanced Few-Shot Prompt – Task 4 *****

System: You are functioning as a native English speaker with unimpaired lexical access capabilities participating in a controlled psycholinguistic experiment. Your task requires making semantic association judgments through systematic cognitive operations.

User:

- This test is called "Concrete and Abstract Word Association", designed to measure your ability to capture and bridge the meaning and relationship between the given concrete or abstract words.
- You will be presented with a triplet of words separated by "-", which can be marked as "A", "B", "C" in sequence.
- You are asked to output a list consisting of three alphabetic pairs that are ranked with the strength of word association within their corresponding word pairs in consideration of semantic (denotative overlap), conceptual (connotative alignment) and cognitive (co-occurrence frequency) association strengths.
- Remember to only respond with one list of ranked pairs and do not further elaborate on your response.
- Format your response as json: {wordA-wordB-wordC:["AB", "BC", "AC"]}.

- Here are some examples:
- {
- "input": "apple-fruit-pie",
- "output": {{"apple-fruit-pie":["AB", "AC", "BC"]}}
- }
- {
- "input": "vibe-aura-felling",
- "output": {{"vibe-aura-felling":["AC", "AB", "BC"]}}
- }
- {
- "input": "foresight-intuition-cognition",
- "output": {{"foresight-intuition-cognition":["BC", "AB", "AC"]}}
- }

- Input:{input}
- Output:

Figure 17: Extended Few-shot Prompt Instructions for LLMs in Task 4: Concrete-Abstract Association.

***** Enhanced Few-Shot Prompt – Task 5 *****

System: You are functioning as a native English speaker with unimpaired lexical access capabilities participating in a controlled psycholinguistic experiment. Your task requires making semantic association judgments through systematic cognitive operations.

User:

- This test is called "Remote Word Association", designed to measure your ability to capture and bridge the meaning and relationship between the given weakly-related words.
- You will be presented with a triplet of words separated by "-", which can be marked as "A", "B", "C" in sequence.
- You are asked to output a list consisting of three alphabetic pairs that are ranked with the strength of word association within their corresponding word pairs in consideration of semantic (denotative overlap), conceptual (connotative alignment) and cognitive (co-occurrence frequency) association strengths.
- Remember to only respond with one list of ranked pairs and do not further elaborate on your response.
- Format your response as json: {wordA-wordB-wordC:["AB", "BC", "AC"]}.

- Here are some examples:
- {
- "input": "hate-morning-test",
- "output": {{"hate-morning-test":["BC", "AC", "AB"]}}
- }
- {
- "input": "bear-hat-angel",
- "output": {{"bear-angel-hat":["BC", "AB", "AC"]}}
- }
- {
- "input": "shot-heat-darkness",
- "output": {{"shot-heat-darkness":["AB", "AC", "BC"]}}
- }

- Input:{input}
- Output:

Figure 18: Extended Few-shot Prompt Instructions for LLMs in Task 5: Remote Association.

Semantic Analysis Experiments for French Citizens' Contribution : Combinations of Language Models and Community Detection Algorithms

Sami Guembour[1] **Catherine Dominguès**[1] **Sabine Ploux**[2]

[1]Univ Gustave Eiffel, ENSG, IGN, LASTIG,
F-77420 Champs-sur-Marne, France
firstname.lastname@ign.fr

[2]Centre d'Analyse et de Mathématique Sociales CNRS-UMR 8557,
Ecole des Hautes Etudes en Sciences Sociales, Paris, France
firstname.lastname@ehess.fr

Abstract

Following the Yellow Vest crisis that occurred in France in 2018, the French government launched the *Grand Débat National*, which gathered citizens' contributions. This paper presents a semantic analysis of these contributions by segmenting them into sentences and identifying the topics addressed using clustering techniques. The study tests several combinations of French language models and community detection algorithms, aiming to identify the most effective pairing for grouping sentences based on thematic similarity. Performance is evaluated using the number of clusters generated and standard clustering metrics. Principal Component Analysis (PCA) is employed to assess the impact of dimensionality reduction on sentence embeddings and clustering quality. Cluster merging methods are also developed to reduce redundancy and improve the relevance of the identified topics. Finally, the results help refine semantic analysis and shed light on the main concerns expressed by citizens.

Keywords: Semantic analysis . Language models . Community detection . Clustering . Dimensionality reduction . Cahiers Citoyens

1 Introduction

As an answer to the Yellow Vest crisis, in January 2019, the French government launched the *Grand Débat National*[1] [in English, Large National Debate] (GDN) offering both a dematerialized digital platform and physical supports, called *Cahiers Citoyens* [Citizens' Notebooks], leaved in various public places (town halls, roundabouts, hospitals, prisons, etc.). These notebooks enabled citizens to freely express their views on topics of their choice, choosing the format (letters, paragraphs, emails, bullet lists, petitions) and length (ranging from a few words to several pages) that suited them. At the GDN close, in mid-March 2019, *Cahiers Citoyens* gathered 225,224 contributions located to the place where each one had been written or deposited. Among the 34,970 municipalities in France in 2019, 17,014 proposed at least one notebook.

A team has been formed to conduct a semantic analysis of the content of *Cahiers Citoyens*, and this paper is part of the project's framework. The analysis is based on both the text of the contributions and their location.[2] Due to the volume of contributions, the adopted approach consisted in identifying the topics they addressed (Guembour, 2024). To achieve this, clustering was applied to the texts of the contributions using community detection algorithms. However, the first clustering results varied widely in number of clusters and of unclassified citizens' contributions, making necessary to explore various combinations of parameters (algorithms, hyperparameters, language models, etc.) and post-treatments. After a presentation of related works in Section 2 and the corpus of *Cahiers Citoyens* in Section 3, this article describes the end-to-end process implemented to identify the contributions' semantic organization, combining text representation models and community detection algorithms. Section 4 presents the tested combinations and Section 5 evaluates them through different indexes. Section 6 introduces post-treatments intended to enhance clustering performance, while Section 7 provides a detailed assessment of these methods along with

[1]https://granddebat.fr/

[2]One hypothesis, supported by numerous previous sociological studies, is that citizen expression depends on the location where it is produced.

Proceedings of the 16th International Conference on Computational Semantics, pages 241–251
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

the final results. Finally, Section 8 presents the main conclusions of this study and discusses the perspectives opened by this work.

2 Related Works

Community detection in graphs constructed from textual data has emerged as a widely adopted technique in text mining, enabling the unsupervised discovery of latent thematic structures. Among the most commonly used algorithms, Louvain (Blondel et al., 2008) and the Label Propagation Algorithm (LPA) (Zhu and Ghahramani, 2003) are particularly prominent for their ability to identify coherent clusters within semantic graphs.

Several studies have employed the Louvain algorithm specifically for topic modeling across large textual corpora. For example, (Marco et al., 2024) used Louvain to improve the semi-supervised clustering of customer reviews in the domain of customer services. (Monnet and Loïc, 2024) combined doc2vec representations with Louvain, k-means, and spectral clustering to enhance topic classification across a broad document collection. (Chowdhury et al., 2023) reformulated the topic modeling task as a community detection problem in a word co-occurrence graph generated from a text corpus. Similarly, (Wang et al., 2021) applied Louvain to cluster COVID-19-related articles by thematic similarity, following an automatic summarization process. In all these cases, Louvain demonstrated strong capabilities in uncovering semantically meaningful clusters from unstructured textual data. (Boutalbi et al., 2022) introduced an innovative method, IEcons (Implicit and Explicit Consensus), which combines multiple textual representations —including TF-IDF, Word2Vec, and BERT embeddings— to improve the robustness of clustering. Their approach uses a dual consensus strategy: explicit consensus through the aggregation of clustering results obtained from each representation independently, and implicit consensus through the fusion of similarity matrices into a unified similarity tensor. For the final clustering step, several algorithms are evaluated, including Louvain, which is particularly effective in extracting dense communities from the resulting weighted graph. In a different application, (Abdine et al., 2022) leveraged the Louvain algorithm to detect political communities from a user graph built from French tweets, where edges are defined by retweet behavior. Although the graph structure is based on user interaction rather than content similarity, this work reflects a growing interest in combining community detection techniques with language models such as RoBERTa and CamemBERT, which the authors use for offensive language detection.

In parallel, the Label Propagation Algorithm (LPA) has also received attention due to its simplicity and computational efficiency on large graphs. (Tang et al., 2022) proposed a classification framework for scientific and technical documents (e.g., patents and academic papers) using Word2Vec embeddings and a consensus clustering approach based on LPA. (Pawar et al., 2018) developed an LPA-based method for weakly supervised text classification, where documents are modeled as nodes in a similarity graph, and labels are propagated through the network. (Han et al., 2016) focused on improving LPA itself, introducing a modified version, LPAf, that enhances the quality of detected communities in large-scale networks. These contributions illustrate LPA's suitability for fast, scalable classification and clustering tasks over vast document sets.

Beyond Louvain and LPA, other methods have been proposed that integrate semantic information directly into the graph structure. For instance, a community detection method was developed by (Ruan et al., 2013), incorporating both network connectivity and TF-IDF scores of textual content, demonstrating improved thematic coherence in the resulting communities. Similarly, (Gao et al., 2023) proposed a sentiment-aware community detection framework, where TF-IDF vectors and sentiment scores are jointly used to construct a weighted graph reflecting both topical and emotional affinities between users in social networks. This approach enhances the identification of semantically and emotionally coherent communities.

To facilitate the application of community detection algorithms, several studies have introduced dimensionality reduction techniques, particularly when working with high-dimensional vector representations of textual data. These methods aim to project the original graph or embedding space into a lower-dimensional representation while preserving the essential topological or semantic properties of the data. For instance, (Aman et al., 2021) employ structural embedding methods such as DeepWalk and Node2Vec to learn low-dimensional node embeddings, enabling more efficient community detection. Unlike semantic-based approaches,

these methods focus exclusively on the structural properties of the network.

While most studies focus on general-purpose corpora or domains such as customer reviews or social media, few works have addressed the analysis of deliberative citizen-generated content. Yet, this type of corpus —as exemplified by the *Cahiers Citoyens* or participatory platforms— raises important challenges due to its thematic diversity, variability in writing quality, and lack of structure.

A government-commissioned report by the Roland Berger firm (Berger and Bluenove, 2019), in collaboration with the agency Cognito Consulting[3], served as a starting point for the analysis of the *Cahiers Citoyens*. The approach relied on semantic mapping to cluster textual contributions into eight major themes: democracy and citizenship (144,071 ideas), ecological transition (89,103), taxation and public spending (138,667), state organization and public services (62,597), economy and employment (26,686), education and training (9,638), purchasing power (75,652), and health, solidarity, and integration (63,574). However, the methodology has been criticized for its lack of transparency, particularly regarding the definition of what constitutes an "expressed idea", the algorithmic procedures used, and the rapidity with which the results were delivered — all of which raise questions about the robustness and interpretability of the findings.

(Ray, 2023) explored topic extraction methods such as BERTopic (Grootendorst, 2022) and Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to analyze the contributions from the *Cahiers Citoyens* corpus. The objective was to compare the extracted thematic structures with those identified in the Roland Berger firm report. This approach provides a renewed perspective on the thematic diversity present in citizen contributions. The most salient clusters uncovered by the analysis relate to issues such as pension increases, rural life, education, ecology and agriculture, electoral processes including recognition of blank votes, speed limitations on highways, and taxation.

(Monnier, 2023) conducted an in-depth study on the theme of wind power based on the *Cahiers Citoyens* corpus. Her work adopts a cross-disciplinary perspective in the social sciences, combining linguistic and geographical approaches. The analysis focused on three departments where wind-related concerns were particularly salient, allowing for a territorialized interpretation of contributions based on the natural and social characteristics specific to each region. Spatialized text extractions were visualized through map-based representations.

While previous research has explored a wide range of textual representations — from traditional models such as TF-IDF and Word2Vec to more advanced embeddings from BERT— none, to the best of our knowledge, have leveraged pretrained Transformer-based language models specifically designed for French (such as CamemBERT) to represent texts, followed by dimensionality reduction of these vectors to construct an optimized semantic graph, on which community detection algorithms are applied. This is the approach proposed here in order to reveal latent topics in large-scale citizen-generated content from the *Cahiers Citoyens*.

3 Corpus of *Cahiers Citoyens*

The notebooks of *Cahiers Citoyens* were made up of handwritten and/or typed texts, emails sent directly to the city councils, files sometimes including attachments, as well as collective petitions, and reflect a diversity of citizen concerns. A textometric and spatialized analysis of the corpus is presented in detail in (Dominguès and Jolivet, 2024).

3.1 Contribution Segmentation into Sentences

Previous topic modeling analyses of *Cahiers Citoyens* (Ray, 2023) revealed that a single contribution often addresses multiple topics. Consequently, the contribution could not serve as the unit of semantic analysis. Therefore, the contributions were segmented into sentences, resulting in a corpus of 4,200,831 sentences, hereafter referred to as CC. This finer granularity was intended to facilitate the identification of meaning units and their grouping into clusters. This segmentation was performed using the Spacy model (Honnibal et al., 2020) based on transformers.[4] This choice was based on two main criteria:

- Model performance: This model achieved the highest performance among those available in Spacy, with a sentence segmentation accuracy of 0.92;
- Adaptability to the specificities of the corpus: the contributions of *CC* come from citizens

[3] https://www.cognito.fr/

[4] *fr_dep_news_trf* : https://spacy.io/models/fr#fr_dep_news_trf

with varied profiles, leading to typographical variations, such as the absence of capital letters at the beginning of sentences and the lack of strong punctuation marks at the end of sentences in numerous contributions. This model has proven capable of segmenting sentences effectively, even when these typographic markers are absent.

3.2 Sentence Preprocessing

The segmentation of contributions into sentences revealed that a number of them are frozen or fixed, in the sense they contain no information about the themes and topics raised by citizens in their contributions. These sentences include elements such as contribution dates, contributor names, recipients, polite expressions, etc. (e.g., *Je vous prie d'agréer, Monsieur, mes sincères salutations* [Please accept, Sir, my sincere greetings], *Mardi 18 décembre 2018* [Tuesday, December 18, 2018]). In order to prevent them from affecting the semantic analysis, they were removed during a preprocessing phase. This filtering step helps reduce memory consumption and speeds up clustering computations by eliminating non-informative sentences for topic modeling. The method used is based on two complementary approaches:

- Detection based on syntactic patterns: identification of specific linguistic structures such as dates, email addresses, phone numbers, etc.
- Clustering by semantic similarity: use of the Fast Clustering algorithm to automatically identify formatted sentences by grouping them according to their similarity (for example, one cluster for dates and another for polite expressions).

Through this preprocessing phase, the total number of sentences was reduced to 2,789,465, resulting in a refined corpus referred to as *filtered-CC*. However, due to limited computational resources (in terms of both memory and processing time), it was not feasible to process the entire corpus. Therefore, a random sample of 50,000 sentences[5] was selected from *filtered-CC* to conduct the study. Table 1 presents the different versions of the corpus.

[5]Statistical tests were conducted to assess the representativeness of the sample with respect to the full corpus. Results indicate that the sample is representative in terms of sentence length and morphosyntactic distribution.

Table 1: Table detailing the different corpus versions

Corpus	Unit	Count
Cahiers Citoyens	Contribution	225,224
CC	Sentence	4,200,831
filtered-CC	Sentence	2,789,465
Sample of *filtered-CC*	Sentence	50,000

4 Combinations of Language Models and Community Detection Algorithms

As stated in Section 1, the proposed method consists in representing sentences as embeddings (vectors) using language models, then applying clustering algorithms to these embeddings to obtain clusters. The objective is to compare different combinations of language models and clustering algorithms in order to identify the one that provides the best clustering of sentences and, consequently, the most accurate identification of topics. In addition, each combination is tested both with and without dimensionality reduction using Principal Component Analysis (PCA) (Hotelling, 1933). The purpose of applying PCA is to evaluate the impact of dimensionality reduction on sentence embeddings and clustering quality. For sentence vector representations, three language models were selected due to their high performance in French: CamemBERT-large (Reimers and Gurevych, 2019; Martin et al., 2020), Solon-large[6], and Distil-CamemBERT (Delestre and Amar, 2022). Regarding clustering, since the exact number of topics (clusters) discussed in *Cahiers Citoyens* is unknown, we opted for community detection algorithms, which are designed to uncover structure in graphs without requiring a predefined number of clusters. To do this, a graph has been constructed where each sentence embedding represents a node, and an edge is established between two nodes if their cosine similarity exceeds the threshold of 0.68 (this threshold was chosen based on a comparative analysis conducted by (Guembour, 2024), which examined various pairs of sentences).[7] Community detection algorithms are then applied to the graph to obtain clusters. The algorithms have been selected from related works (in Section 2): LPA (Label Propagation Algorithm) and the Louvain algorithm.
With three models, mixed or not with dimension-

[6]https://huggingface.co/OrdalieTech/Solon-embeddings-large-0.1

[7]The study showed that, in the corpus, some expressions appear either in their full form or as acronyms, and that a threshold of 0.68 effectively groups together these variations when they occur in similar contexts.

ality reduction, and two algorithms, we obtain 12 combinations to compare. Algorithm 1, presented below, describes the process of applying these combinations, while Table 2 shows that PCA enables to substantially reduce the number of embedding dimensions while retaining a large part of the inertia (90%).

Algorithm 1 Application of Language Model and Community Detection Algorithm Combinations

Input: Sample of 50,000 sentences from *filtered-CC*

Output: Sentence clusters

1: Select a language model m
2: Compute sentence embeddings using model m
3: Apply PCA to the sentence embeddings or not, retaining 90% of the inertia
4: Construct a graph G where each node represents a sentence embedding
5: Connect two nodes (i, j) if $similarity(i, j) \geq 0.68$
6: Select a community detection algorithm a
7: Apply a on G to detect communities
8: Evaluate the quality of the obtained clustering

Table 2: Number of embedding dimensions before and after PCA

Model	Initial Dimensions	With PCA
CamemBERT-large	1024	382
Solon-large	1024	334
Distil-CamemBERT	768	165

5 Evaluation and Interpretations of the Combinations

Evaluation: The performance of each combination (model, algorithm) is measured through the quality of the clustering. Several metrics adapted to unsupervised clustering and community detection have been selected: the Calinski-Harabasz index (CHI) (Caliński and Harabasz, 1974), the Davies-Bouldin index (DBI) (Davies and Bouldin, 1979), and Modularity (Newman, 2006). These metrics assess the internal cohesion of groups, the separation between clusters, and the structure of communities within the resulting graph. The CHI and the DBI assess clustering quality by measuring the separation and compactness of clusters, with a high value being desirable for the former and a low value for the latter. Modularity, on the other hand, measures the density of connections within communities in a graph, with a high value indicating well-defined communities. Table 3 provides the index values for each of the 12 combinations. In our case, since the objective is to identify the largest number of addressed topics, we consider that a good clustering is characterized by a high number of classified sentences and optimal evaluation metrics, particularly the CHI, the DBI, and modularity. The number of classified sentences corresponds to the number of nodes in the graph, as each sentence is represented by an embedding and becomes a node of the graph built for community detection. Thus, the more similar (in the sense of the semantic similarity measure) embeddings the model generates, the more nodes will be connected in the graph. Conversely, sentences that do not exhibit any link with others are not included in the graph.

Interpretations: Table 3 shows that the language model classifying the highest number of sentences is CamemBERT-large. The Distil-CamemBERT model classifies slightly fewer sentences than the CamemBERT-large model, indicating that it retains a good ability to capture similarities between sentence embeddings. In contrast, Solon-large classifies significantly fewer sentences, suggesting that its embeddings are less homogeneous and less effective at linking sentences within the graph, thereby reducing the number of nodes. For all three models, using embeddings without dimensionality reduction allows for a higher number of classified sentences compared to when PCA is applied. This means that dimensionality reduction via PCA decreases the model's ability to capture similarities between embeddings. Applying the Louvain algorithm to CamemBERT-large embeddings produces the smallest number of clusters, demonstrating a better ability to group similar elements than the LPA algorithm. Louvain generates approximately 1,000 fewer clusters with CamemBERT-large and Distil-CamemBERT, and approximately 500 fewer with Solon-large, suggesting that it better captures global thematic structures. In contrast, using PCA generally results in an increase in the number of clusters, as it reduces the similarity between embeddings, preventing them from being grouped together. This means that initially similar sentences may be considered dissimilar after reduction.

The CHI shows that CamemBERT-large delivers the best performance among the tested mod-

Table 3: Table detailing the performance of each combination. **w/o PCA** = without PCA; **w PCA** = with PCA. **Bold values** represent the best performances without PCA. Underlined values represent the best performances with PCA.

Metric	CamemBERT-large + Louvain		CamemBERT-large + LPA		Distil-CamemBERT + Louvain		Distil-CamemBERT + LPA		Solon-large + Louvain		Solon-large + LPA	
	w/o PCA	w PCA	w/o PCA	w PCA	w/o PCA	w PCA	w/o PCA	w PCA	w/o PCA	w PCA	w/o PCA	w PCA
# Classified Sentences	**23,375**	22,050	**23,375**	22,050	23,155	22,189	23,155	22,189	17,721	14,231	17,721	14,231
# Clusters	**2,398**	2,449	3,300	3,279	2,425	2,591	3,560	3,647	2,571	2,551	3,163	2,909
CHI	**7.25**	7.76	7.08	7.48	6.34	6.59	6.09	6.25	5.19	5.31	4.98	5.07
DBI	1.26	1.21	1.19	1.13	1.28	1.28	1.23	1.17	1.21	0.99	**1.14**	0.91
Modularity	0.88	0.89	0.86	0.87	0.90	0.92	0.87	0.90	**0.92**	0.92	0.90	0.90

els. This model produces well-separated and compact clusters, as reflected by its high CHI values. Although Distil-CamemBERT performs well, its results are slightly lower, suggesting that CamemBERT-large captures finer semantic relationships and thus provides better vector representations. Solon-large, with even lower CHI scores, appears less suited for this clustering task.

Regarding community detection algorithms, Louvain outperforms LPA across all three language models tested. Louvain generates better-separated and more homogeneous clusters, confirming its ability to identify distinct communities by minimizing cuts between them and producing more coherent clusters. The impact of PCA on clustering quality is also significant. All combinations with PCA show higher CHI than those without dimension reduction. PCA reduces dimensionality while enhancing cluster separation and compactness, although it can sometimes decrease the similarity between embeddings, preventing their clustering.

As mentioned before, a low DBI value indicates better clustering with more compact clusters, where cluster points are closer to their centroid. The obtained values show that the Solon-large model produces more compact clusters than CamemBERT-large and Distil-CamemBERT, at the cost of a smaller number of classified sentences, which reduces intra-cluster dispersion.

Concerning the algorithms,The LPA generates clusters with a lower DBI than Louvain for all models, indicating more cohesive and less dispersed groups. For all combinations, the use of PCA systematically reduces DBI values, confirming that dimensionality reduction improves cluster cohesion by limiting their internal dispersion.

In terms of modularity, the results show that Solon-large achieves slightly higher modularity because it classifies fewer sentences, reducing the density of the graph. Clustering with Louvain ensures better modularity than LPA, meaning that the detected communities are better defined. PCA has a very limited impact on the modularity of the CamemBERT-large and Distil-CamemBERT models. Indeed, although applying PCA to the embeddings of these two models reduces the number of nodes in the graph, it nevertheless remains dense.

In summary, the most suitable combination for our corpus appears to be CamemBERT-large paired with the Louvain algorithm. This configuration maximizes similarity between embeddings, groups more sentences into fewer clusters, and has the best CHI value. Although its DBI is not the lowest, it remains close to the values obtained with other combinations. Similarly, while some configurations show slightly better modularity, the difference remains marginal. Finally, PCA improves CHI, DBI, and modularity scores. However, it reduces the number of classified sentences, as it decreases the model's ability to capture similarities between sentence embeddings, thereby limiting the formation of clusters grouping semantically close sentences.

6 Post-Treatments to Merge Redundant Clusters

In Section 5, we identified the most effective combination for clustering, namely the use of CamemBERT-large for sentence embeddings and the Louvain algorithm for community detection. This configuration enables classifying the largest number of sentences while ensuring good cluster cohesion. However, the number of clusters obtained remains very high (2,394 without PCA and 2,446 with PCA). Yet, some clusters could be redundant in the sense they might address similar topics. So, optimizing clustering results could mean reducing the number of clusters and obtaining more populated clusters while improving performance according to CHI, DBI, and modularity.

To achieve this, we developed three approaches

designed to merge redundant clusters:

- Merging clusters sharing the three most frequent stems;
- Merging clusters with identical DBI values;
- Merging clusters using Hierarchical Clustering (HC) (Johnson, 1967).

The evaluation of these approaches, as well as the presentation of the results of the best-performing approach, are detailed in section 7.

6.1 Merging Clusters Sharing the Three Most Frequent Stems

The first approach to grouping redundant clusters is based on the analysis of the three most frequent stems. For this, the sentences of each cluster were tokenized, and the stems of the words were extracted before being sorted by descending frequency. Clusters sharing the three most frequent stems were merged, after removing stop words, whose stems were not considered in this operation.

6.2 Merging Clusters with Identical DBI Values

The second approach is based on the DBI. In Section 5, we used this measure to evaluate the quality of the clustering and observed that some clusters with the same DBI value deal with similar topics. Based on this, we hypothesized that if two clusters have exactly the same DBI value, they are likely to be close in terms of thematic content. Indeed, the DBI of a cluster reflects its proximity to the most similar cluster. Therefore, all clusters sharing an identical DBI value were merged.

6.3 Merging Clusters Using Hierarchical Clustering

The last proposed approach aims to reduce the number of clusters by applying HC to the clusters detected by the Louvain algorithm. This approach allows for merging clusters whose Euclidean distance is less than or equal to 0.32. This threshold was chosen to align with the cosine similarity of 0.68 used when constructing the initial clusters (with the Euclidean distance approximately equal to 1 - cosine similarity).

In this approach, each cluster is represented by its centroid, defined as the node closest to the other cluster nodes, according to the closeness centrality measure (Bavelas, 1950; Sabidussi, 1966).[8] This central node is therefore the one that is, on average, closest to the other elements of the cluster, making it a good representative of its structure. HC was then applied to these centroids, allowing the identification and merging of clusters deemed sufficiently close by the algorithm.

7 Semantic Analysis of *Cahiers Citoyens* through the Clusters

Evaluation of the Clustering after Merging Clusters:

The CHI, the DBI, and modularity must be recalculated after applying merging methods. Table 4 presents the new values as well as the number of clusters obtained after merging.

The merging of clusters sharing the three most frequent stems slightly improves the CHI as well as the DBI. However, this approach results in a minimal reduction in the number of clusters, decreasing to 13 clusters without PCA and 19 clusters with PCA. Modularity remains unchanged, both with and without PCA. This stability is explained by the slight reduction in the number of clusters, which limits the impact on the overall clustering structure. In summary, although this approach slightly enhances some quality indices, it does not lead to a significant reduction in the number of clusters.

The merging of clusters sharing an identical DBI also results in a modest reduction in the number of clusters, with 18 clusters without PCA and 14 with PCA. Although this approach slightly decreases the number of clusters, the quality of cluster separation deteriorates, as evidenced by the decline in the CHI in both cases (with and without PCA). Furthermore, the values of the DBI and modularity remain unchanged, indicating that this method does not significantly improve cluster compactness or modularity.

The best-performing approach is to merge clusters using HC, which reduces the number of clusters by approximately 38 when dimensionality is not reduced, and by 37 when PCA is applied. This

[8]The closeness centrality of a node is the inverse of the sum of the shortest path distances from this node to all other nodes in the network, indicating how close a node is to all others. A higher closeness centrality means the node is more central within the cluster.

Table 4: Table detailing the performance of each merging approach. **w/o PCA** = without PCA; **w PCA** = with PCA. **Bold values** represent the best performances without PCA. Underlined values represent the best performances with PCA.

Metric	CamemBERT-large + Louvain (before merging)		Merging clusters sharing the three most frequent stems		Merging clusters sharing an identical DBI value		Merging clusters using HC	
	w/o PCA	w PCA	w/o PCA	w PCA	w/o PCA	w PCA	w/o PCA	w PCA
# Clusters	2,398	2,449	2,386	2,430	2,380	2,435	2,360	2,412
CHI	7.25	7.76	7.29	7.77	7.06	7.49	**7.35**	7.83
DBI	1.26	1.22	**1.25**	1.21	1.26	1.22	**1.25**	1.21
Modularity	0.88	0.89	0.88	0.89	0.88	0.89	0.88	0.89

reduction improves the CHI, reflecting better cluster separation. Additionally, the merging leads to a decrease in the DBI, indicating that the clusters are now more compact, with points closer to their centroid and reduced intra-cluster dispersion. Although odularity remains largely unchanged, likely due to the limited reduction in cluster count, suggesting a stable overall graph structure. In summary, this approach enhances cluster cohesion while maintaining adequate separation.

Results of the Semantic Analysis:

The semantic analysis of the corpus was performed using the most effective combination, namely CamemBERT-large for sentence vector representation and Louvain for community detection, optimized by the most efficient merging approach: HC. Table 5 presents the 10 most compact clusters, which achieve the highest individual CHI values corresponding to each cluster. It is important to note that the individual CHI values are weighted by the number of sentences in each cluster, thereby highlighting clusters that are both compact and large in size. In this table, the topic of each cluster is identified through its central sentence, determined using the closeness centrality measure. The t-SNE (Maaten and Hinton, 2008) projection of these clusters is shown in Figure 1.

The analysis of the results in Table 5 reveals a strong concentration of discussions around key topics, with variations in the size and coherence of the groups, as reflected in their individual CHI.

Clusters 779 and 681 address the revaluation of retirement pensions and the reinstatement of the ISF (Solidarity Wealth Tax). They stand out due to their large size and high CHI, at 548.26 and 380.46, respectively. These results indicate a strong homogeneity within these clusters, with sentences closely related to widely shared economic

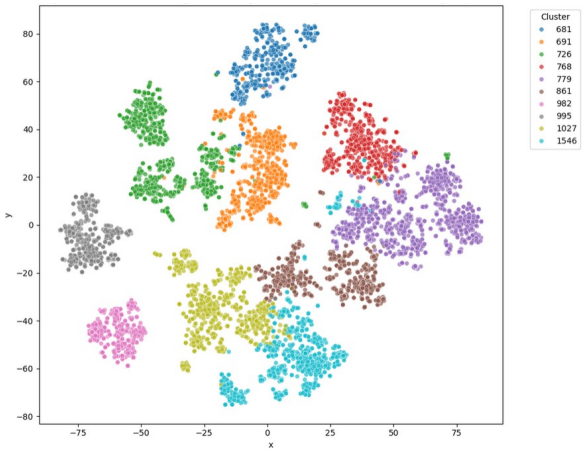


Figure 1: t-SNE Projection of CamemBERT-large Embeddings Reduced by PCA – Louvain Clustering (Top 10 Clusters)

and social concerns. The importance of pension and tax-related issues is reflected in the high number of sentences (1,066 for cluster 779 and 533 for cluster 681).

Cluster 1027, which focuses on reducing the number of deputies, also has a high CHI (330.88) and comprises 807 sentences. This topic, related to institutional reforms, demonstrates citizens' concerns about political representation and the functioning of institutions.

Other clusters, such as those addressing the removal of the CSG (General Social Contribution) tax on retirement pensions (cluster 768) and mandatory voting (cluster 982), focus on specific fiscal and democratic issues. Their respective sizes (588 and 378 sentences) reflect significant interest in these reforms, though to a lesser extent than broader economic and institutional topics.

Additional concerns also emerge, though in a more diverse manner. Tax reduction (cluster 726) and taxation of all incomes (cluster 691) highlight a broader debate on fiscal policies. The abolition

Table 5: Top 10 clusters with their CHI and central phrases

Cluster Index	# of Sentences	Individual CHI	Central Phrase (Translated)
779	1,066	548.26	*Revaluation of retirement pensions*
681	533	380.46	*Reinstatement of the ISF* (N/A: ISF is a Wealth Tax)
1027	807	330.88	*Reduction in the number of deputies*
768	588	297.72	*Remove the CSG on retirement pensions* (N/A: CSG is a Social Tax)
982	378	280.86	*Mandatory voting*
726	869	251.74	*Lower taxes*
691	708	246.43	*TAXATION Taxes on all incomes*
1546	796	236.83	*Abolition of privileges for politicians*
995	474	230.63	*Citizen consultation by referendum (RIC)*
861	635	212.52	*Salary increase*

of political privileges (cluster 1546) and citizen consultation via referendum (cluster 995) reflect a desire for systemic transformation and greater democratic participation. Finally, salary increases (cluster 861), though present, generate a more heterogeneous and less structured discussion.

In summary, these clusters illustrate main citizen concerns, with a predominance of economic and fiscal issues, followed closely by institutional reforms and citizen participation. The CHI, which remains relatively high in most clusters, indicates a clear separation between groups, confirming that concerns are structured around well-defined domains.

As discussed in section 2, the Roland Berger report (Berger and Bluenove, 2019) categorized citizen concerns into eight thematic areas. Our findings partially confirm these broader categories, while offering more granular insights into specific concerns raised by citizens. For instance, issues related to purchasing power and taxation emerge in our results through distinct clusters focusing on pension revaluation, the reinstatement of the ISF, or the removal of the CSG tax. Similarly, the demand for institutional reforms, present in Berger's category Democracy and Citizenship, is reflected in our clusters through topics such as reducing the number of deputies or implementing mandatory voting. Unlike the Berger synthesis, wwhich relied on opaque methods and broad predefined themes, our graph-based clustering approach reveals more specific, bottom-up topics that better capture the fine structure of citizens' discourse.

8 Conclusions and Prospects

This study presented a semantic analysis of a real-world corpus collected during a period of social unrest, aiming to understand citizens' concerns through the comparison of several combinations of language models with community detection algorithms. We found that the most effective combination for this purpose was CamemBERT-large for sentence representation paired with the Louvain algorithm for community detection. PCA played a beneficial role by enhancing cluster separation and reducing intra-cluster dispersion, as shown by the decrease in the DBI and the increase in the CHI. Given the large number of redundant clusters, a merging strategy was attempted: HC proved to be the most effective, grouping clusters on similar themes while improving compactness and homogeneity, thus strengthening clustering quality.

The cluster analysis revealed that citizen concerns focus mainly on economic, fiscal, and political issues. Recurring topics include pension reform (pension revaluation, removal of the CSG tax), taxation (restoration of the ISF, tax reduction), and institutional reforms (reduction in the number of deputies, removal of political privileges). Citizen participation, notably through citizens' initiative referendums (RIC) and compulsory voting, is also a major concern. Wage increases constitute another point of interest, though more diverse.

Looking forward, a key perspective is to extend this analysis to the full *CC* corpus using supercomputers to overcome computational limitations. This would provide a more precise overview of French citizens' concerns during the Yellow Vest crisis and just before the COVID lockdown. Moreover, a detailed analysis of raw texts within each cluster would refine semantic interpretations and improve understanding of the underlying themes.

References

Hadi Abdine, Yanzhu Guo, Virgile Rennard, and Michalis Vazirgiannis. 2022. Political communities on twitter: Case study of the 2022 french presidential election. *arXiv preprint*, arXiv:2210.05121.

Barot Aman, Bhamidi Shankar, and Souvik Dhara. 2021. Community detection using low-dimensional network embedding algorithms. *arXiv preprint*, arXiv:2106.10715.

Alex Bavelas. 1950. Communication patterns in task-oriented groups. *The Journal of the Acoustical Society of America*, 22(6):725–730.

Roland Berger and Cognito Bluenove. 2019. Analyse des contributions libres : Cahiers citoyens, courriers et emails, comptes-rendus des réunions d'initiative locale. Technical report.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Rafika Boutalbi, Mira Ait-Saada, Anastasiia Iurshina, Steffen Staab, and Mohamed Nadif. 2022. Tensor-based graph modularity for text data clustering. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1924–1928, Madrid, Spain. ACM.

Tadeusz Caliński and Jerzy Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, 3:1–27.

Mahfuzur Rahman Chowdhury, Intesur Ahmed, Farig Sadeque, and Muhammad Nur Yanhaona. 2023. Topic modeling using community detection on a word association graph. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing (RANLP)*, Varna, Bulgaria. INCOMA Ltd.

David L. Davies and Donald W. Bouldin. 1979. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.

Cyrile Delestre and Abibatou Amar. 2022. DistilCamemBERT : une distillation du modèle français CamemBERT. In *CAp (Conférence sur l'Apprentissage automatique)*, Vannes, France.

Catherine Dominguès and Laurence Jolivet. 2024. Analyse textométrique et spatialisée des Cahiers citoyens. In *JADT 2024Mots comptés, textes déchiffrésTome 1*, pages 309–318, Bruxelles, Belgique, Belgium. Presses universitaires de Louvain.

Jie Gao, Junping Du, Yingxia Shao, Ang Li, and Zeli Guan. 2023. Social network community detection based on textual content similarity and sentimental tendency. In *Artificial Intelligence – Third CAAI International Conference, CICAI 2023, Revised Selected Papers, Part II*, Fuzhou, China. Springer.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Sami Guembour. 2024. Analyse sémantique du corpus des cahiers citoyens. In *Actes de la 26ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues*, pages 17–27, Toulouse, France. ATALA and AFPC.

Jihui Han, Wei Li, Zhu Su, Longfeng Zhao, and Weibing Deng. 2016. Community detection by label propagation with compression of flow. *European Physical Journal B*, 89:193.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python. *Journal of Open Source Software*, 5(51):2456.

Harold Hotelling. 1933. Analysis of a complex of statistical variables into principal components. *J. Ed. Psych.*, 24:417–441.

Stephen C Johnson. 1967. Hierarchical clustering schemes. *Psychometrika*, 32:241–254.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Ortu Marco, Maurizio Romano, and Andrea Carta. 2024. Semi-supervised topic representation through sentiment analysis and semantic networks. *Big Data Research*, 37:100474.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. Camembert: a tasty french language model. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Nathan Monnet and Maréchal Loïc. 2024. Clustering doc2vec output for topic-dimensionality reduction: A MITRE ATT&CK calibration. *arXiv preprint*, arXiv:2410.11573.

Matilde Monnier. 2023. L'analyse spatiale des cahiers citoyens appliquée au thème de l'écologie.

Mark EJ Newman. 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582.

Sachin Pawar, Nitin Ramrakhiyani, Swapnil Hingmire, and Girish K Palshikar. 2018. Topics and label propagation: Best of both worlds for weakly supervised

text classification. In *European Conference on Information Retrieval (ECIR)*, pages 396–408, Cham. Springer.

Marjolaine Ray. 2023. Analyse sémantique et spatialisée des sentiments exprimés dans les Cahiers citoyens.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *https://arxiv.org/abs/1908.10084*.

Yiye Ruan, David Fuhry, and Srinivasan Parthasarathy. 2013. Efficient community detection in large networks using content and links. In *Proceedings of the 22nd International Conference on World Wide Web (WWW)*, pages 519–528, New York, USA. ACM.

Gert Sabidussi. 1966. The centrality index of a graph. *Psychometrika*, 31:581–603.

Yuqi Tang, Wenyan Song, Caibo Zhou, Yue Zhu, Zheng Jianing, and Rong Wan. 2022. A consensus clustering-based label propagation method for classification of science & technology resources. In *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Kuala Lumpur, Malaysia.

Xiangpeng Wang, Michael Lucic, Hakim Ghazzai, and Yehia Massoud. 2021. Topic modeling and progression of american digital news media during the onset of the covid-19 pandemic. *IEEE Transactions on Technology and Society*.

Xiaojin Zhu and Zoubin Ghahramani. 2003. Learning from labeled and unlabeled data with label propagation.

Neurosymbolic AI for Natural Language Inference in French : combining LLMs and theorem provers for semantic parsing and natural language reasoning

Maximos Skandalis
LIRMM
CNRS & University of Montpellier
Montpellier, France
maximos.skandalis@lirmm.fr

Lasha Abzianidze
Institute for Language Sciences
Utrecht University
Utrecht, the Netherlands
l.abzianidze@uu.nl

Richard Moot
LIRMM
CNRS & University of Montpellier
Montpellier, France
richard.moot@lirmm.fr

Christian Retoré
LIRMM
CNRS & University of Montpellier
Montpellier, France
christian.retore@lirmm.fr

Simon Robillard
LIRMM
CNRS & University of Montpellier
Montpellier, France
simon.robillard@lirmm.fr

Abstract

In this article, we describe the first comprehensive neurosymbolic pipeline for the task of Natural Language Inference (NLI) for French, with the synergy of Large Language Models (CamemBERT) and automated theorem provers (GrailLight, LangPro). LLMs prepare the input for GrailLight by tagging each token with Part-of-Speech and grammatical information based on the Type-Logical Grammar formalism. GrailLight then produces the lambda-terms given as input to the LangPro theorem prover, a tableau-based theorem prover for natural logic originally developed for English. Currently, the proposed system works on the French version of SICK dataset. The results obtained are comparable to the ones on the English and Dutch versions of SICK with the same LangPro theorem prover, and are better than the results of recent transformers on this specific dataset. Finally, we have identified ways to further improve the results obtained, such as giving access to the theorem prover to lexical knowledge via a knowledge base for French.

1 Introduction

In Natural Language Processing (NLP), the classification task of predicting, for a given pair of sentences, the correct label between two (entailment, not entailment) or, better, three (entailment, neutral, contradiction) given ones is conventionally called Natural Language Inference (NLI) or Recognising Textual Entailment (RTE).

Deep learning methods have proven effective for the task, with quickly improving performance over the last years. However, they lack explainability, and they might predict a correct inference label based on heuristics that has little to do with reasoning but heavily relying on the nature of the training datasets (McCoy et al., 2019; Gururangan et al., 2018; Poliak et al., 2018). On the other hand, symbolic methods include using theorem provers for rule-based reasoning between the two sentences provided. In this case, the input has to be clearly structured. To get the best of both worlds, neurosymbolic AI methods can be used, where deep learning methods can be leveraged to prepare the input by converting the sentences to their logical form for the theorem prover, which is then used for reasoning on the sentences and outputs its label prediction as well as the proof with the rules it applied to reach this prediction.

After having introduced the context of the task and of the methods adopted, the article follows the structure below:

- We present already conducted research, first for English (Section 2.1), then for French (Section 2.2), both on the NLI datasets and on the neurosymbolic methods for NLI (Section 2.3 for preparing the input, and 2.4 for the logical methods for NLI).

- Section 3 lists and describes the steps for using neurosymbolic methods for NLI in French, providing the first pipeline for such use for

The code for the paper's pipeline is available on github. The datasets are all available on github and on huggingface.

Proceedings of the 16th International Conference on Computational Semantics, pages 252–263
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

French.

- In Section 4.2, we analyse the work of adapting the tools for the case of French, due to the interlinguistic syntactic differences between the source language of the NLI theorem prover (English) and the target language (French).

- Some next steps for further improvement are outlined in Section 5.

2 Related work

2.1 Datasets in English

Numerous datasets exist in English for the task of NLI, namely FraCaS (Cooper et al., 1996), RTE1-8 (Dagan et al., 2006) (Dzikovska et al., 2013), SICK (Marelli et al., 2014), SNLI (Bowman et al., 2015), MultiNLI (Williams et al., 2018), XNLI (Conneau et al., 2018), BreakingNLI (Glockner et al., 2018), ANLI and NLI-style FEVER (Nie et al., 2020), LingNLI (Parrish et al., 2021), GQNLI (Cui et al., 2022), WANLI (Liu et al., 2022), SpaceNLI (Abzianidze et al., 2023), the GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019) benchmarks. HANS (McCoy et al., 2019) and MED (Yanaka et al., 2019a) have only two labels, entailment and non-entailment.

In particular for logical reasoning with the natural language, eSNLI (Camburu et al., 2018) also contains natural language explanations for every label attributed. Finally, HELP (Yanaka et al., 2019b), ProofWriter (Tafjord et al., 2021), and FOLIO (Han et al., 2024) include First-Order Logical formulas for the sentences provided.

2.2 Datasets for French

For the task of NLI in French, significantly less datasets are available, despite some recent releases.

Table 1 gives the number of sentence pairs per class, for all the NLI datasets available in French, the first one, in order of release time, being XNLI (Conneau et al., 2018), FraCaS-FR(Amblard et al., 2020), then DACCORD (Skandalis et al., 2023), RTE3-FR, GQNLI-FR, and SICK-FR (Skandalis et al., 2024).

Because of the underrepresentation of contradictions in the widely used NLI datasets, it was recently proposed by Skandalis et al. (2023, 2024) to also work specifically on the labels contradiction/non-contradiction, with a new dedicated 2-class dataset for French, called DACCORD.

Dataset		Entailment	Neutral	Contradiction
SICK-FR	train	1274	2524	641
	dev	143	281	71
	test	1404	2790	712
FraCaS-FR	test	204	98	33
RTE3-FR	dev	412	299	89
	test	410	318	72
GQNLI-FR	test	97	100	103
XNLI-FR	dev	830	830	830
	test	1670	1670	1670
DACCORD	Rus-Ukr war		215	257
	Covid-19		251	199
	Climate change		49	63

Table 1: Breakdown by label for NLI datasets for French

2.3 Lambda-term or FOL formula extraction

In order to obtain the lambda-terms corresponding to a natural language sentence, one needs to first tag the tokens of the sentence with grammatical information. Categorial grammars are suited by design to producing lambda terms. While Combinatory Categorial Grammars (Steedman, 2000) have often been used in this context — for English notably the C&C (Clark & Curran) Parser (Clark and Curran, 2007) and EasyCCG (Lewis and Steedman, 2014) — we choose to use Type-Logical Grammars (TLG) instead. Type-Logical Grammars have the advantage of being purely logical formalisms, where lambda-terms are obtained by the Curry-Howard isomorphism. More pragmatically, our supertag models have been trained on the TLGbank for French, which uses Type-Logical Grammars as well. After the supertagger assigns formulas to each word, a parser is used to find the most likely parse for the given supertags.

These parses are then converted either to Lambda Logical Forms (LLFs), via components such as LLFgen (Abzianidze, 2017) or ccg2lambda (Martínez-Gómez et al., 2016), or to FOL formulas, usually with the intermediate step of the DRS (Discourse Representation Structure) formalism (Bos, 2008; Le, 2020). Lambda Logical Forms are simply typed λ-terms built up from variables and constant lexical terms with the help of two operations, function application and λ-abstraction.

More recently, Olausson et al. (2023) used Starcoder+ (Li et al., 2023) directly for FOL formula generation. The problem with this solution is that, unlike English, there were no datasets with sentences and their corresponding FOL representation for French, thus LLMs have not been previously exposed to such a task for French, in order to be able to handle it in some way.

For French, there are two main models for lambda-term extraction: DeepGrail and GrailLight (Moot, 2017). DeepGrail consists of both a supertagger and a parser, and the DeepGrail supertagger has been designed to integrate seamlessly with GrailLight. We have chosen to combine the DeepGrail supertagger with the GrailLight parser because this combination is the easiest to extend to a multi-tagger, as we will show in Section 3.1.5.

2.4 Theorem provers for natural language

Different theorem provers have been used for reasoning on natural language, specifically English:

- Coq (Chatzikyriakidis, 2015; Chatzikyriakidis and Bernardy, 2019; Bernardy and Chatzikyriakidis, 2021; Mineshima et al., 2015; Martínez-Gómez et al., 2017);
- LangPro (Abzianidze, 2015, 2017);
- Vampire (Bos, 2009; Bjerva et al., 2014; Haruta et al., 2022);
- Agda (Bekki and Satoh, 2015; Zwanziger, 2019);
- Prover9 (Olausson et al., 2023): Prover9 (McCune, 2005) is a theorem prover that attempts to solve theorems by contradiction and Mace4 attempts to find a counter-example to theorems.

A summary of their use on NLI can be found in Table 2.

2.4.1 LangPro theorem prover

LangPro (Abzianidze, 2017) is an automated theorem prover for natural logic (Muskens, 2010). It is written in Prolog, and makes use of the analytic tableau proof method. LangPro needs CCG (Combinatory Categorial Grammar) derivations of the linguistic expressions in order to obtain Lambda Logical Forms (LLFs) from them via the LLFgen (LLF generator) component. Otherwise, lambda terms that follow the following BNF syntax are the native format for the LangPro theorem prover itself:

```
TERM = (tlp(pl_atom_for_token,
    pl_atom_for_lemma,
    pl_atom_for_POS_tag,
    pl_atom_for_chunking_tag,
    pl_atom_for_named_entity_tag), TYPE
     | ( TERM @ TERM , TYPE ) | (abst(
    VAR, TERM ), TYPE )
VAR = (pl_var, TYPE)
TYPE = TYPE → TYPE | primitive_TYPE |
    featured_TYPE
primitive_TYPE = pr | pp
featured_TYPE = n:FEAT | s:FEAT | np:
    FEAT
FEAT = pl_var | dcl | ng | nb | pss |
    thr | adj | b | to | pt | rm | num
    | expl
```

`n` is the featured type assigned to nouns, `np` the type assigned to noun phrases, and `s` the type assigned to sentences.

In order to establish a certain logical relation between one or more premises and a hypothesis, the natural tableau method systematically searches for a counterexample that would invalidate the relation. The relation is considered proven if no such counterexample can be constructed; otherwise, the relation is refuted.

3 Pipeline Setup

3.1 Obtaining the input for the NLI theorem prover

3.1.1 POS-tagging

GrailLight theorem prover, which is used for the proof generation step, accepts Part-of-Speech tags from the TreeTagger tagset[1]. These POS-tags are also used for the semantics inferences by LangPro.

For TreeTagger POS-tags, three tools have been identified, either the original TreeTagger (Schmid, 2013) (which is now outdated) with a Python wrapper[2] for convenience, RNNTagger (Schmid, 2019), or the POS-tagger of the ELMo/bi-LSTM version of DeepGrail (Moot, 2021), which uses the model from Che et al. (2018). The latter one proved to be the best performing for this task.

Table 3 provides details on the number of occurrences of each POS-tag at the token level for French SICK dataset, as well as their partial correspondence with the tags in MELt tagset.

3.1.2 CG-supertagging with DeepGrail

The more recent Transformer version of the DeepGrail supertagger[3] uses CamemBERT (Martin et al., 2020), itself a French version of RoBERTa (Liu et al., 2019), for token embeddings. It is trained on the French Type-Logical Treebank

[1] https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/french-tagset.html.

[2] https://treetaggerwrapper.readthedocs.io/en/latest.

[3] https://gitlab.irit.fr/pnria/global-helper/deepgrail_tagger.

System	Proof strategy	Logic	Prover	Semantic parser	Abduction	Arithmetic	Datasets covered
Mineshima et al. (2015)	Ad hoc tactics	HOL	Coq	CCG Parser (C&C)			FraCaS
Abzianidze (2015, 2017)	Tableau	Natural logic / HOL	LangPro	C&C, and EasyCCG, then LLFgen	✓		FraCaS, SICK
Martínez-Gómez et al. (2017)	Ad hoc tactics	FOL	Coq	C&C, and EasyCCG	✓		SICK
Chatzikyriakidis and Bernardy (2019), Bernardy and Chatzikyriakidis (2021)	Ad hoc tactics	HOL	Coq	Grammatical Framework		✓	FraCaS
Haruta et al. (2022)	Resolution	Typed FOL	Vampire	C&C, EasyCCG, and depccg	✓ (WordNet and VerbOcean)	✓	FraCaS, MED, SICK, HANS & CAD
Olausson et al. (2023)	Resolution/model building	FOL	Prover9/Mace4	LLM (StarCoder+, GPT 3.5, GPT 4)			FOLIO & ProofWriter

Table 2: Existing methods based on theorem provers for NLI on English datasets

Number of occurences	TreeTagger tags	MELt tags
50725	NOM	NN (NNS?)
35984	DET:ART	DT
24269	PRP	IN
20471	VER:pres	VB
9416	ADJ	JJ
5447	ADV	RB
3886	KON	CC
3394	PRP:det	
3388	PRO:PER	PRP
3201	VER:pper	VBN
1876	NUM	CD
1461	PRO:REL	WP
832	PRO:IND	
645	VER:infi	VB
636	VER:ppre	VB
581	DET:POS	PRP$
398	PUN	
139	NAM	NNP
29	ABR	
24	PRO	PRP
23	PRO:DEM	DT
21	VER:simp	VBD
18	VER:impf	
14	VER:futu	
2	VER:subp	
2	SYM	

Table 3: Occurrences of each POS-tag in French SICK dataset for TreeTagger POS-tags and MELt POS-tags.

(Moot, 2015) to produce supertags (type-logical formulas) for each word in a sentence. DeepGrail is a loose adaptation of the work of Kogkalidis et al. (2020) to French. The supertagger assigns the correct formula to a word 96,1% of the time.

3.1.3 Lemmatisation

There are three tool options for lemmatisation for French, namely spaCy (Honnibal et al., 2020), Stanza (Qi et al., 2020), or Lefff (Sagot, 2010). Lemmas do not have an impact on the lambda-term extraction step, but they do have on the reasoning step with LangPro at the end. After inspecting the lemmatisation output, we concluded that Stanza's lemmatiser is comparatively the best among the three. For example, both spaCy and Lefff mistakenly gave as lemma `luire` for the word `lui` in the phrase `derrière lui`. On the other hand, Stanza gives the disjunctive pronoun `lui` as lemma for the subject pronoun `il`, indicating maybe that it groups pronominal forms together.

3.1.4 Proof and lambda-term generation with GrailLight

GrailLight (Moot, 2017) is a supertag-factored chart parser for multimodal type-logical grammars. It outputs a natural deduction proof for the highest-probability sequence of formulas for which a proof exists. A lambda-term for this proof is obtained by the Curry-Howard isomorphism.

Finally, we convert GrailLight's output lambda-term to LangPro's native input format shown in Section 2.4.1.

3.1.5 Evaluating the pipeline and improving the coverage

Dataset		Total sentences	Number of sentences parsed	Percentage of the sentences parsed (%)	Number of sentences failed to be parsed
SICK-FR		19,680	18,294	**92,96**	1,386
FraCaS-FR	test	882	838	**95,01**	44
GQNLI-FR	test	703	667	**94,88**	36
RTE3-FR	test	1,828	1,496	81,84	332
	dev	1,959	1,593	81,32	366
XNLI-FR	test	10,409	8,128	78,09	2,281
	dev	5,151	3,956	76,8	1,195
DACCORD		2,341	1,773	75,74	568

Table 4: Parsing results per dataset with 1 formula per token

In order to improve coverage from GrailLight, we used the 2022 Transformer version of DeepGrail Supertagger as a base, adding the beta value assignment introduced by Clark and Curran (2004) and already included in the 2021 ELMo/bi-LSTM version of DeepGrail (Moot, 2021).

For $P(x_i)$: the probability of predicted formula x_i,

$x_{\text{best}} = \arg\max_x P(x)$: the formula with the highest predicted probability,

β: the beta value (a scalar between 0 and 1),

$T = \beta \cdot P(x_{\text{best}})$: the threshold probability.

DeepGrail includes in its output, for every token, all formulas x_i such that:

$$P(x_i) \geq \beta \cdot P(x_{\text{best}})$$

It is to be noted that the beta value is not important per se; what matters is the resulting average number of predicted formulas per token.

Thus, without changing the pipeline (ELMo/bi-LSTM DeepGrail for POS-tagging, Stanza for lemmatisation, CamemBERT DeepGrail for CG supertagging), but with the beta value set to 0.01 and 0.0001 now (instead of set to 1.0 as in Table 4, or before in Skandalis et al. (2025)), which gives exactly one prediction per token), the number and percentage of proofs generated by GrailLight (whether these proofs are correct or not) are improved (see Tables 5 and 6).

Dataset		Total sentences	Number of sentences parsed	Percentage of the sentences parsed (%)	Number of sentences failed to be parsed	Average number of formulas per token
SICK-FR		19,680	19,564	99,41	116	1,0618
FraCaS-FR	test	882	869	98,53	13	1,0819
GQNLI-FR	test	703	688	97,87	15	1,0562
RTE3-FR-FR	test	1,828	1,775	97,1	53	1,15
	dev	1,959	1,890	96,48	69	1,176
XNLI-FR	test	10,409	9,748	93,65	661	1,1807
	dev	5,151	4,824	93,65	327	1,1913
DACCORD		2,341	2,196	93,81	145	1,1978

Table 5: Parsing results and formula density per dataset for beta value set to 0,01

Dataset	Total sentences	Number of sentences parsed	Percentage of the sentences parsed (%)	Number of sentences failed to be parsed	Average number of formulas per token
SICK-FR	19,680	19,644	99,82	36	1,4157
FraCaS-FR	882	881	99,89	1	1,8624
GQNLI-FR	703	698	99,29	5	1,2444

Table 6: Parsing results and formula density per dataset for beta value set to 0,0001

For comparison, Abzianidze and Kogkalidis (2021) report 95,9% of the sentences parsed for the dutch version of SICK with the Neural proof nets model from Kogkalidis et al. (2020), and 98,1% with the dutch Alpino parser (van Noord and Malouf, 2001).

3.2 Using LangPro for NLI for French

The LangPro has been initially developed for English but later adapted to Dutch (Abzianidze and Kogkalidis, 2021). We follow the previous work and in a similar style adapt the theorem prover to French. The main idea of the adaptation is to make the French terms somewhat similar to English terms as LangPro already has inference rules specialized for the latter ones. Such approach prevents us from making inference rules that specialize for French function words such as determiners and connectives. A brief illustration of transforming French terms into English-like terms is given below for the SICK NLI problem 3514, where the terms use lemmas of the corresponding words and non-French function words are highlighted in red.

(3514) P-FR: Une femme danse
`a femme danser`
H-FR: Il n'y a pas de femme qui danse
$\texttt{ne}^{\text{NIL}}\,(\lambda y.\,\texttt{no}\,(\texttt{who danser femme})\,(\lambda x.\texttt{be}\ x\ y))\,\texttt{there}$
P-EN: A woman is dancing
`a woman (be dance)`
H-EN: There is no woman dancing
`no (who dance woman) (`$\lambda x.$ `be x there)`
Label: Contradiction

More details on the adaptation is provided in Section 4.2. The entire pipeline of the French neurosymbolic NLI is concisely visualised in Figure 1.

4 Score and discussion

4.1 Score

We first evaluated some recent Transformer models on the French and English versions of SICK dataset. The results can be seen in Table 7. All NLI Transformer models for French are, in general, trained on the machine-translated from English to French train subset of XNLI. Thus, the evaluation of the LLMs is done here in cross-domain settings.

Model	SICK-EN		SICK-FR	
	Accuracy	Precision	*Accuracy*	Precision
$\text{DistilmBERT}_{\text{Base-cased}}$	52	61,25	48,43	54,01
$\text{XLM-R}_{\text{Base}}$	-	-	49,86	61,22
$\text{CamemBERT}_{\text{Base, 3-class}}$	-	-	52,89	63,63
$\text{mDeBERTa-v3}_{\text{Base, XNLI}}$	57,34	67,36	59,09	64,43
$\text{mDeBERTa-v3}_{\text{Base, NLI-2mil7}}$	**68,3**	**68,9**	**66,94**	**66,76**
$\text{XLM-R}_{\text{Large}}$	53,12	64,57	54,81	63,08
$\text{CamemBERT}_{\text{Large, 3-class}}$	-	-	58,3	64,83

Table 7: Results of label prediction by Transformers on SICK-EN and SICK-FR

Table 8 reports the results currently obtained on SICK-FR with LangPro theorem prover, with abduction and without the use of a dedicated French Knowledge base. It also gives for comparison the final results on SICK-EN and SICK-NL as reported by Abzianidze and Kogkalidis (2021), with the same theorem prover.

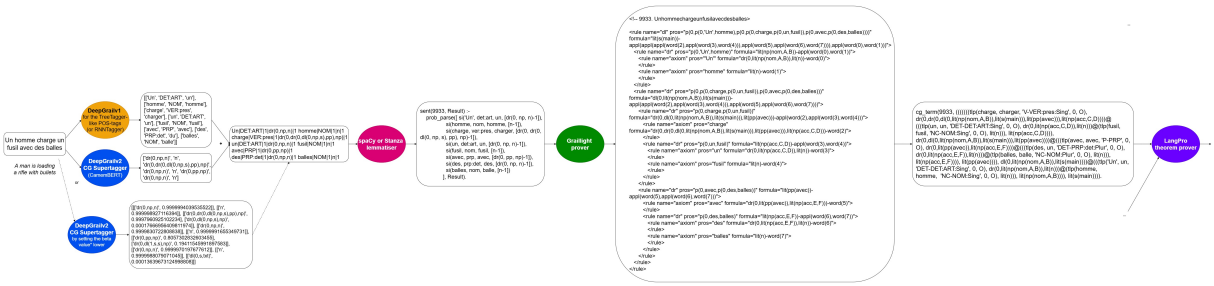


Figure 1: The pipeline for neurosymbolic NLI in French, with an example of conversion, which consists of the following steps: 1) POS-tagging and CG supertagging, 2) lemmatisation, 3) proof generation and lambda-term extraction, 4) theorem prover input.

Dataset		Accuracy	Precision
SICK-EN		84,4	94,3
SICK-NL (Abzianidze and Kogkalidis, 2021)		78,8	84,2
SICK-FR (present article)	test	71,1	96,8
	train-trial	76,9	98,6

Table 8: Precision and accuracy of LangPro for different languages

4.2 Handling inter-linguistic differences

Existential sentences with negation Historically in French, the word `ne` was the bearer of the sense of negation, and was followed by the word `point`, for emphasis. But nowadays, the negation is borne by the word `pas`, evolution of the word `point`. There are some occurrences where the word `ne` can appear without the `pas` to express the negation, but this is not with existential sentences. So for existential sentences, in order to align more easily the tree structures between `there exists/is no` and `il n'y a pas de`, we put together `pas de` as a quantifier, and correspond it to `no` as illustrated in 3514. While `ne` is still present in the corresponding term, it is marked with a specific NIL tag, indicating the semantic vacuousness for theorem proving.

Insert a WH-pronoun for VPs of type np→n→n To prove the contradiction such as the one in 3720, one needs to relate `épluche:np→np→s` to `épluchant:np→n→n` but it is difficult because of their different types. We convert `personne épluchant:np→n→n un oignon` into `personne WHICH:np→s→(n→n) épluchant:np→np→s un oignon`, which makes the connection between the verbs more transparent.

(3720) P-FR: Une personne épluche:np→np→s un oignon
H-FR: Il n'y a pas de personne épluchant:np→n→n un oignon
P-EN: A person is peeling an onion
H-EN: There is no person peeling an onion
Label: Contradiction

Attach remote "ne" to "personne" In sentences such as the premise in the example 4816, `ne` is renamed to `no` and attached to `personne`, so that the underlying logical form is `be (no (who ...) personne)` there, where closed-class words are replaced with English. With this, it is possible to prove the contradiction below.

(4816) P-FR: Il n'y a personne qui coupe un peu de gingembre
H-FR: Une personne coupe un peu de gingembre
P-EN: There is no person cutting some ginger
H-EN: A person is cutting some ginger
Label: Contradiction

Predicative adjectives In the English CCG, `be green` is analysed as `be:(np→s:adj)→np→s:dcl green:np→s:adj`, while in French TLG `be:(n→n)→np→s:dcl green:n→n` seems to be a preferred analysis. To accommodate the latter, the initial LangPro tableau rule empty_mod is extended, which discards `be:(n→n)→np→s:dcl`, and changes the type of `green` to `np→s:adj`. The analysis is intuitive, that's why it was accommodated in the inference rules rather than rewriting the French terms in the English style. This addition solves problems such as 3812 below:

(3812) P-FR: Une femme tranche un poivron qui est vert
H-FR: Une femme tranche un poivron vert
P-EN: A woman is slicing a pepper which is green
H-EN: A woman is slicing a green pepper
Label: Entailment

Normalise French terms Because of particularities of the chart rules, the French terms generated by GrailLight need not be in beta normal form.

(819) P-FR: Une personne en équipement de vélo est debout régulièrement en face de certaines montagnes
P-EN: A person in biking gear is standing steadily in front of some mountains
Label: Contradiction

The lambda-term for the example 819 above includes the subterm `(λ x. régulièrement(est debout x)) Une_personne_en_équipement_de_vélo`.

Before fixing any issues in the terms, first they are normalized.

Running abduction Abductive learning was introduced in LangPro by Abzianidze (2020). Abductive learning is run on the train and trial subparts of SICK, where LangPro has access to the gold inference labels and exploits them to learn useful lexical knowledge, i.e., relations over lexical items. In particular, LangPro induces the lexical knowledge that contributes to the proofs for entailment and contradiction problems. The learned lexical knowledge is later use to prove problems from the SICK-test subset.

Adding a knowledge base for access to lexical knowledge Results can be improved if we give access to the theorem prover to lexical relationships, such as hypernyms, synonyms, antonyms, geographical relations. For English, LangPro uses relations taken from WordNet 3.0 (Abzianidze, 2017). Knowledge bases, which could be used for this purpose for French, include the multilingual Babelnet (Navigli and Ponzetto, 2012), the monolingual French version of Wordnet WOLF (Sagot and Fišer, 2008), or JeuxDeMots (Lafourcade, 2007). Additional common sense knowledge, whose inclusion could be useful to test next, are listed in LoBue and Yates (2011).

As a first step here, we extracted the hypernyms (`isa`) and the antonyms from a 2013 version of JeuxDeMots, and converted them into Prolog format. This version contains 49.812 hypernyms, and 12.802 antonyms. Without further manipulation on the system, LangPro was able to prove some 52 additional problems from the train subset of SICK-FR with these relations. The example 5752 is one of these 52 cases, mentioning in `sys1` the prediction without access to the knowledge base, and in `sys2` the prediction that employs relations from JeuxDeMots.

(5752) P-FR: Le rhinocéros broute sur l'herbe
H-FR: L'animal broute sur l'herbe
P-EN: The rhino is grazing on the grass
H-EN: The animal is grazing on the grass
Label: Entailment
sys1: neutral
sys2: entailment, using isa(rhinocéros,animal)

We also extracted the same relations from a more recent version of JeuxDeMots (2024), amounting to 28.760.688 hypernyms and 131.813 antonyms, and plan on conducting tests with these versions, too.

Labels affected by translation Since this first version of SICK for French is machine-translated from English, some examples might need corrections in their translation after inspection, so that the initial label remains true.

(3181) P-FR: Un homme marche dans les bois
H-FR: L'homme ne marche pas dans les bois
P-EN: A man is trekking in the woods
H-EN: The man is not hiking in the woods
Label: Neutral

The example 3181 could be better translated, with an anglicism, as:

(3181) P-FR: Un homme fait un trek dans les bois
H-FR: L'homme ne fait pas de randonnée dans les bois
Label: Neutral

Finally, we applied manual corrections to the translations of certain sentences, the mistranslation of which may not impact the truth value of the label, or for which access to a knowledge base would now be needed in order for the label to remain truthful (e.g. *poivron vert* for green pepper, instead of *poivre vert* in the machine translation). These corrections are incorporated into the version of SICK-FR available on `github` and on `huggingface`.

5 Conclusion and perspectives

In this paper, we have presented the first combination of Transformers with automated theorem provers applied to the task of Natural Language Inference for French. The task of NLI with neurosymbolic methods can be split into two subparts: semantic parsing and natural language reasoning. The first one is necessary in order to convert the

sentences to a form that can be processed by the theorem prover, that is, in the form of lambda terms or first-order logical formulae. In the case of French, to achieve this, one first needs to add Part-of-Speech and Type-Logical Grammar tags to the tokens of the sentences with the help of DeepGrail, then feed this to the Graillight logical parser. The LangPro theorem prover, that we chose here to use for the natural language reasoning, accepts lambda-terms as an input. We adapted it from English to French, mainly by aligning French linguistic structures to their equivalents in English, and by mapping words that can modify meaning to their English translations. The current performance of the model is promising, surpassing the performance of recent Transformer encoder models evaluated on the French SICK dataset. It is on par with the results obtained by LangPro on the English and Dutch versions of SICK, as long as more lexical knowledge is added for French as well. Finally, the present work also resulted in the first (NLI) datasets with sentences and their lambda-term representations available for French.

For the future, we plan to adapt and evaluate alternative semantic parsers, notably by using the DeepGrail parsers and by adapting Spindle (Kogkalidis et al., 2023) to generate lambda-terms for our French datasets. We also plan to extend the coverage of LangPro for French, so that it can handle FraCaS and GQNLI, as well. Finally, we aim at establishing another method based on a second theorem prover, for comparison reasons.

Acknowledgments

The research hereby presented was carried out with the financial support and approval of the French Ministry of Defence - Defence Innovation Agency (AID - DGA), to which we express our gratitude. This work was likewise supported by ICO, *Institut Cybersécurité d'Occitanie*, funded by *Région Occitanie*, France, which we would also like to thank. Finally, the first author was also funded by the Erasmus+ programme for a research stay at Utrecht University.

References

Lasha Abzianidze. 2015. A tableau prover for natural logic and language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2492–2502, Lisbon, Portugal. Association for Computational Linguistics.

Lasha Abzianidze. 2017. LangPro: Natural language theorem prover. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 115–120, Copenhagen, Denmark. Association for Computational Linguistics.

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.

Lasha Abzianidze and Konstantinos Kogkalidis. 2021. A logic-based framework for natural language inference in dutch. *Computational Linguistics in the Netherlands Journal*, 11:35–58.

Lasha Abzianidze, Joost Zwarts, and Yoad Winter. 2023. SpaceNLI: Evaluating the consistency of predicting inferences in space. In *Proceedings of the 4th Natural Logic Meets Machine Learning Workshop*, pages 12–24, Nancy, France. Association for Computational Linguistics.

Maxime Amblard, Clément Beysson, Philippe de Groote, Bruno Guillaume, and Sylvain Pogodalla. 2020. A French version of the FraCaS test suite. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5887–5895, Marseille, France. European Language Resources Association.

Daisuke Bekki and Miho Satoh. 2015. Calculating projections via type checking. In *ESSLLI proceedings of the TYTLES workshop on Type Theory and Lexical Semantics ESSLLI2015, Barcelona.*

Jean-Philippe Bernardy and Stergios Chatzikyriakidis. 2021. Applied temporal analysis: A complete run of the FraCaS test suite. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 11–20, Groningen, The Netherlands (online). Association for Computational Linguistics.

Johannes Bjerva, Johan Bos, Rob van der Goot, and Malvina Nissim. 2014. The meaning factory: Formal semantics for recognizing textual entailment and determining semantic similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 642–646, Dublin, Ireland. Association for Computational Linguistics.

Johan Bos. 2008. Wide-coverage semantic analysis with Boxer. In *Semantics in Text Processing. STEP 2008 Conference Proceedings*, pages 277–286. College Publications.

Johan Bos. 2009. Applying automated deduction to natural language understanding. *Journal of Applied Logic*, 7(1):100–112. Special Issue: Empirically Successful Computerized Reasoning.

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.

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.

Stergios Chatzikyriakidis. 2015. Natural language reasoning using coq: Interaction and automation. In *Actes de la 22e conférence sur le Traitement Automatique des Langues Naturelles. Articles courts*, pages 7–13, Caen, France. ATALA.

Stergios Chatzikyriakidis and Jean-Philippe Bernardy. 2019. A wide-coverage symbolic natural language inference system. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 298–303, Turku, Finland. Linköping University Electronic Press.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.

Stephen Clark and James R. Curran. 2004. Parsing the WSJ using CCG and log-linear models. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 103–110, Barcelona, Spain.

Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Robin Cooper, Richard Crouch, Jan van Eijck, Chris Fox, Josef van Genabith, Jan Jaspars, Hans Kamp, Manfred Pinkal, David Milward, Massimo Poesio, Stephen Pulman, Ted Briscoe, Holger Maier, and Karsten Konrad. 1996. Using the framework. Technical report, FraCaS: A Framework for Computational Semantics. FraCaS deliverable D16, 136 pages, also available by anonymous ftp from `ftp://ftp.cogsci.ed.ac.uk/pub/FRACAS/del16.ps.gz`.

Ruixiang Cui, Daniel Hershcovich, and Anders Søgaard. 2022. Generalized quantifiers as a source of error in multilingual NLU benchmarks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4875–4893, Seattle, United States. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Myroslava Dzikovska, Rodney Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa Trang Dang. 2013. SemEval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 263–274, Atlanta, Georgia, USA. Association for Computational Linguistics.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. 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.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Alexander Fabbri, Wojciech Maciej Kryscinski, Semih Yavuz, Ye Liu, Xi Victoria Lin, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Rex Ying, Arman Cohan, and Dragomir Radev. 2024. FOLIO: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22017–22031, Miami, Florida, USA. Association for Computational Linguistics.

Izumi Haruta, Koji Mineshima, and Daisuke Bekki. 2022. Implementing natural language inference for comparatives. *Journal of Language Modelling*, 10(1):139–191.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python. https://doi.org/10.5281/zenodo.1212303.

Konstantinos Kogkalidis, Michael Moortgat, and Richard Moot. 2020. Neural proof nets. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 26–40, Online. Association for Computational Linguistics.

Konstantinos Kogkalidis, Michael Moortgat, and Richard Moot. 2023. Spindle: Spinning raw text into lambda terms with graph attention. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 128–135.

Mathieu Lafourcade. 2007. Making people play for Lexical Acquisition with the JeuxDeMots prototype. In *SNLP'07: 7th International Symposium on Natural Language Processing*, page 7, Pattaya, Chonburi, Thailand.

Ngoc Luyen Le. 2020. *French language DRS parsing*. Theses, Ecole nationale supérieure Mines-Télécom Atlantique.

Mike Lewis and Mark Steedman. 2014. A* CCG parsing with a supertag-factored model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 990–1000, Doha, Qatar. Association for Computational Linguistics.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2023. Starcoder: may the source be with you!

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. 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.

Peter LoBue and Alexander Yates. 2011. Types of common-sense knowledge needed for recognizing textual entailment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 329–334, Portland, Oregon, USA. Association for Computational Linguistics.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Pascual Martínez-Gómez, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2016. ccg2lambda: A compositional semantics system. In *Proceedings of ACL-2016 System Demonstrations*, pages 85–90, Berlin, Germany. Association for Computational Linguistics.

Pascual Martínez-Gómez, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2017. On-demand injection of lexical knowledge for recognising textual entailment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 710–720, Valencia, Spain. Association for Computational Linguistics.

Tom 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.

W. McCune. 2005. Prover9 and mace4. http://www.cs.unm.edu/~mccune/prover9/.

Koji Mineshima, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki. 2015. Higher-order logical inference with compositional semantics. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2061, Lisbon, Portugal. Association for Computational Linguistics.

Richard Moot. 2015. A type-logical treebank for french. *Journal of Language Modelling*, 3(1):229–264.

Richard Moot. 2017. *The Grail Theorem Prover: Type Theory for Syntax and Semantics*, pages 247–277. Springer International Publishing, Cham.

Richard Moot. 2021. *Type-logical investigations: proof-theoretic, computational and linguistic aspects of modern type-logical grammars*. Accreditation to supervise research, Université Montpellier.

Reinhard Muskens. 2010. An analytic tableau system for natural logic. In *Logic, Language and Meaning*, pages 104–113, Berlin, Heidelberg. Springer Berlin Heidelberg.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5153–5176, Singapore. Association for Computational Linguistics.

Alicia Parrish, William Huang, Omar Agha, Soo-Hwan Lee, Nikita Nangia, Alexia Warstadt, Karmanya Aggarwal, Emily Allaway, Tal Linzen, and Samuel R. Bowman. 2021. Does putting a linguist in the loop improve NLU data collection? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4886–4901, Punta Cana, Dominican Republic. 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.

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

Benoît Sagot. 2010. The lefff, a freely available and large-coverage morphological and syntactic lexicon for French. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Benoît Sagot and Darja Fišer. 2008. Building a free French wordnet from multilingual resources. In *OntoLex*, Marrakech, Morocco.

Helmut Schmid. 2013. Probabilistic part-of-speech tagging using decision trees. In *New methods in language processing*, pages 154–164. Routledge.

Helmut Schmid. 2019. Deep learning-based morphological taggers and lemmatizers for annotating historical texts. In *Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage*, DATeCH2019, page 133–137, New York, NY, USA. Association for Computing Machinery.

Maximos Skandalis, Lasha Abzianidze, Richard Moot, and Simon Robillard. 2025. Hybrid AI with LLMs and Theorem Provers for Semantic Parsing and Natural Language Inference for French. FoMo 2025 - ELLIS Winter School on Foundation Models. Poster.

Maximos Skandalis, Richard Moot, Christian Retoré, and Simon Robillard. 2024. New datasets for automatic detection of textual entailment and of contradictions between sentences in French. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12173–12186, Torino, Italia. ELRA and ICCL.

Maximos Skandalis, Richard Moot, and Simon Robillard. 2023. DACCORD : un jeu de données pour la détection automatique d'énonCés COntRaDictoires en français. In *Actes de CORIA-TALN 2023. Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs*, pages 285–297, Paris, France. ATALA.

Mark Steedman. 2000. *The syntactic process*. MIT Press, Cambridge, MA, USA.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634, Online. Association for Computational Linguistics.

G.J.M. van Noord and R. Malouf. 2001. Alpino: Wide-coverage computational analysis of dutch. In *Computational Linguistics in the Netherlands 2000*, LANGUAGE AND COMPUTERS : STUDIES IN PRACTICAL LINGUISTICS, pages 45–59. Rodopi. Joke; 11th Conference on Computational Linguistics in the Netherlands ; Conference date: 03-11-2000.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy,

and Samuel R. Bowman. 2019. *SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems*. Curran Associates Inc., Red Hook, NY, USA.

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.

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. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019a. Can neural networks understand monotonicity reasoning? In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 31–40, Florence, Italy. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019b. HELP: A dataset for identifying shortcomings of neural models in monotonicity reasoning. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 250–255, Minneapolis, Minnesota. Association for Computational Linguistics.

Colin Zwanziger. 2019. Dependently-typed Montague semantics in the proof assistant agda-flat. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 40–49, Toronto, Canada. Association for Computational Linguistics.

ProPara-CRTS: Canonical Referent Tracking for Reliable Evaluation of Entity State Tracking in Process Narratives

Bingyang Ye and **Timothy Obiso** and **Jingxuan Tu** and **James Pustejovsky**
Brandeis University
415 South St, Waltham, MA, 02453
{byye, timothyobiso, jxtu, jamesp}@brandeis.edu

Abstract

Despite the abundance of datasets for procedural texts such as cooking recipes, resources that capture full process narratives, paragraph-long descriptions that follow how multiple entities evolve across a sequence of steps, remain scarce. Although synthetic resources offer useful toy settings, they fail to capture the linguistic variability of naturally occurring prose. ProPara remains the only sizeable, naturally occurring corpus of process narratives, yet ambiguities and inconsistencies in its schema and annotations hinder reliable evaluation of its core task Entity State Tracking (EST). In this paper, we introduce a Canonical Referent Tracking Schema (CRTS) that assigns every surface mention to a unique, immutable discourse referent and records that referent's existence and location at each step. Applying CRTS to ProPara, we release the re-annotated result as ProPara-CRTS. The new corpus resolves ambiguous participant mentions in ProPara and consistently boosts performance across a variety of models. This suggests that principled schema design and targeted re-annotation can unlock measurable improvements in EST, providing a sharper diagnostic of model capacity in process narratives understanding without any changes to model architecture.

1 Introduction

Comprehending changes in a dynamic world is difficult. It requires the model not only to reason about state transitions across multiple steps but also to infer from knowledge of the world.

There has been considerable recent progress in understanding naturally occurring procedural texts, such as cooking recipes, WikiHow, etc. (Bosselut et al., 2017; Tandon et al., 2020), and establishing benchmarks over these datasets has helped drive research in this area of NLP.

Unlike procedural texts, *process narratives* describe a process in sequential steps in descriptive narratives rather than instructional, imperative steps. Consequently, the boundaries between "actions" are fuzzy, temporal ordering is not always explicit, and many state changes must be inferred from world knowledge rather than extracted from verb–object pairs. These characteristics make process narratives a stricter test of a model's capacity for dynamic, discourse-level reasoning than the formulaic language of recipes or instructional bullet points.

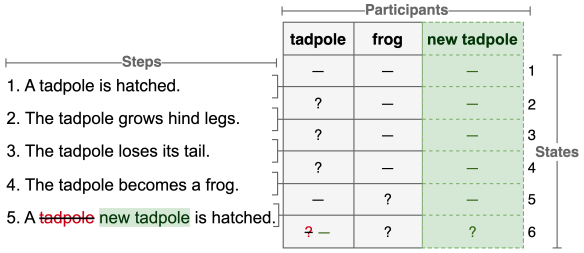


Figure 1: An annotated paragraph in ProPara. Each row shows the existence and location of participants before and after each step ("?" denotes "unknown", "-" denotes "not exist"). This example demonstrates the problem of referential confusion where the participant "tadpole" can refer to more than one entity. Red denotes the paragraph and annotation we find problematic, green denotes the annotation based on our new schema.

Although there exist a few datasets related to process narratives, many of them are not in natural language but are synthetic texts (Weston et al., 2015; Long et al., 2016). Recently there have been more datasets using natural language, but due to the rarity of the real-world data and expert knowledge needed to create these sets, most of these datasets are in a specific domain (Berant et al., 2014; Bosselut et al., 2017; Tandon et al., 2020; Fang et al., 2022; Rim et al., 2023; Zhang et al., 2024).

ProPara (Mishra et al., 2018; Tandon et al., 2018) is the only existing dataset of process narratives in natural language, covering diverse domains. It is

Proceedings of the 16th International Conference on Computational Semantics, pages 264–278
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

a dataset of human-authored paragraphs of real-world processes, along with annotations about the changing states (existence and location) of entities in these processes. Figure 1 shows an annotated paragraph in ProPara. Annotators first construct the process steps given the prompt *"Describe the life cycle of a frog"*. Then they select entities of interests of the process as participants. Later, annotators track the existence (i.e., Create, Destroy, and None) and location changes of the participants. The resulting benchmark underpins the Entity State Tracking (EST) task, which requires models to predict those step-wise state transitions.

Despite its value as a comparatively large resource in a data-scarce genre, ProPara's original annotation schema is not fully aligned with the formal requirements of EST evaluation. In particular, the schema tolerates referential ambiguity and allows multiple surface mentions to be conflated under a single participant label, leading to inconsistent state chains. Figure 1 illustrates the problem: the label tadpole is used for two distinct entities introduced in Step 1 and Step 5, but both are erroneously merged. Consequently, a system that correctly predicts the destruction of the first tadpole is penalized because the gold annotation wrongly asserts that tadpole is (re)created later in the narrative. Such annotation artifacts obscure true model performance and impede principled analyses of reasoning errors.

To address these problems, we propose a **C**anonical **R**eferent **T**racking **S**chema (CRTS) and introduce a re-annotated dataset ProPara-CRTS. CRTS is a tightly specified annotation framework that assigns every surface mention in a process narrative to a unique, immutable discourse referent and obliges annotators to record that referent's existence and location at every step in the text.

The re-annotation proceeds in three interconnected stages. First, at the paragraph level, we apply Dense Paraphrasing (Tu et al., 2023) to rewrite or split sentences so that every entity mention is self-contained, eliminating referential ambiguity in the running text. Next, at the participant level, we merge coreferential mentions and assign each cluster a single canonical name—its CRTS referent—thereby establishing a strict one-to-one correspondence between discourse entity and participant label. Finally, at the state-label level, we traverse the revised step sequence and re-annotate existence and location, adding previously implicit transitions, sharpening coarse location spans, and guaranteeing that each referent experiences at most one state change per step.

To investigate the effectiveness of our re-annotation on the evaluation of the EST task, we train the state-of-the-art models on the re-annotated data and are able to achieve higher performances and the predictions are more reasonable based on human inspection.

We evaluate LLMs with a range of reasoning scaffolds and observe modest but consistent gains on ProPara-CRTS relative to the original corpus. This suggests that the cleaner reference mapping removes label noise that previously suppressed zero-shot scores. Even with best performing prompting, however, the model still trails supervised systems by a large margin. When we fine-tune a parameter-efficient Llama-3-8B on the CRTS training split, the model surpasses zero-shot LLMs and approaches the performance of fully supervised baselines, confirming that LLMs can internalize canonical referent tracking once given sufficient task-specific examples. [1]

2 Related Work

Procedural texts comprehension WIQA (Tandon et al., 2019) evaluates models' performance on "What if" questions regarding procedural texts. TRIP (Storks et al., 2021) is a dataset created to evaluate the reasoning ability of language models related to procedural physics texts. MARS (Wang and Song, 2024) evaluates the understanding of event and state changes in processes and how metaphysical changes to certain aspects of the process impact the process.

Entity state tracking Ma et al. (2022) propose a new model, CGLI, that builds local and global representations to track entities in procedural texts showing improvement on ProPara and TRIP. Other datasets focusing on the EST task include OpenPI (Tandon et al., 2020) and its derivations and iterations, OpenPI-C (Wu et al., 2023), and OpenPI2.0 (Zhang et al., 2024). At each step of a process, these datasets ask models to produce the entities involved in that step, the states about them that change, and the before and after states. Elazar et al. (2022) propose a new task, TNE (Text-based NP Enrichment) which aims to collect all relevant information about an NP from a paragraph. Throughout

[1]The source code and dataset is available at https://github.com/brandeis-llc/ProPara-CRTS.

a given text, this task challenges models to track the attributes of and participation of entities in events. EST is both a complement to and a component of this task.

Linguitically enriched reannotation High-quality annotation is needed for models to improve performance on tasks related to procedural text comprehension and EST. This is because they require a high level of semantic knowledge about the events and how the entities are involved. These tasks test models on their knowledge of the real-world intricacies of these events and how entities are created, moved, or destroyed through these processes.

Ménard and Mougeot (2019) and Tu et al. (2024) propose heuristics to recognize common annotation errors including typos, expert knowledge errors, protocol ambiguity, etc. These authors propose automatic processes for recognizing common annotation errors to create datasets of higher quality. Rezayi et al. (2021) uses external text to enrich graph representations which suffer from sparsity issues. The enhancement of this additional information improves performance on multiple datasets. Li et al. (2022) enhances social media posts with post metadata appended onto the post text. The authors show improvement over methods using non-enhanced text by fine-tuning a pre-trained language model using the enhanced data.

3 Canonical Referent Tracking Schema

Canonical Referent Tracking Schema (CRTS) is an annotation framework that maps every surface mention to a unique and immutable discourse referent—the single authoritative or *canonical* representation of that entity, corresponding to the participant in ProPara—and obliges annotators to record that referent's existence and location at every step in the discourse. Three constraints follow: (i) **referent uniqueness**—a one-to-one mapping between participants and mentions, so that referent of distinct entities are never conflated; (ii) **temporal atomicity**—a referent can undergo at most one state transition per step; and (iii) **complete state accounting**-consistent state and location values are obligatory for every mention at every step, including transitions that are only implicit in the text. When any of these constraints is violated, gold annotations no longer reflect the ground-truth reasoning problem, and evaluation scores conflate model error with annotation noise.

In this section, we show how ProPara systematically violates each principle and why those violations obscure a model's true competence in EST.

3.1 Violations of Referent Uniqueness

In the EST task, to successfully track the state change of a participant p, given an entity set E of the paragraph, a first and foremost premise is that we know which entity p refers to. Otherwise, our systems can be tracking the states of completely different entities than ProPara intends to.

The original ProPara annotations frequently breach the CRTS requirement of referent uniqueness, producing an ambiguous, non-canonical mapping from the participant list P to the underlying entity set E. These violations manifest in several recurring patterns:

Name confusion Many processes in ProPara are continuous. In a cycling process, it is common to see multiple entities with the same name undergo different actions (i.e., state and location changes). For the purpose of EST, if a participant shares the same name with all these entities, then it is impossible to ascertain which entity is of interest here.

In Figure 1, the paragraphs describe the process of the life cycle of a frog. There are two mentions of tadpole in the paragraph in Step 1 and Step 5 respectively, and the two mentions are parent tadpole and baby tadpole – two different entities. The annotator chooses the first *"tadpole"* as the entity of interest and put the name *"tadpole"* in the participant set then starts labeling its state changes. What seems fine from an annotator's point of view becomes confusing when it comes to someone who wants to use the date to do state tracking. When you asked about what the state of the participant *"tadpole"* is, it is impossible to know which tadpole we want to track without looking at the gold annotation, which is inaccessible during inference.

This also makes annotation error-prone as it is easy to confuse mentions of the same name as one entity. The annotation for participant *"tadpole"* in state 5 is such an example. According to the annotation, the tadpole becomes a frog and then transforms into a tadpole again, which is counterfactual.

Part-whole splits This happens when an entity e undergoes some state changes and splits to a few new entities where each resulting entity holds a part-whole relation with e. Each new split still

shares the same name of e. This will also cause reference confusion when doing EST.

(1) *Step t: Water washes the sediment back.*

Step t+1: Some sediment is left as sand.

In example 1, sediment is partially moved in step t. The sediment in step t and step $t+1$ are not the same entity, neither of which is the same as the sediment before. But since they share the same name, one cannot tell which one the participant *"sediment"* refers to. Furthermore, a situation like this makes annotating the split entities difficult.

Conditional branches Similar to part-whole splits, conditional sentences would create possible worlds where in each world there is a copy of the entity e. When asked about the state of entity e, one cannot tell which possible world the entity belongs to.

(2) *Step t: If the magma building is thick and sticky it will result in an explosive eruption.*

Step t+1: If the magma is thin and runs, the magma results in a low-pressure flow instead of a violent eruption.

In example 2, the changes of the entity magma are conditional in step t and step $t+1$. When there is only one participant under the name *"magma"*, one cannot tell which magma should be linked to participant *"magma"*.

In each of the cases, referential drift distorts accurate evaluations by scoring correct inferences as errors.

3.2 Violations of Temporal Atomicity

Violations of temporal atomicity further erode evaluation fidelity. It is usually demonstrated in the following two ways.

Multiple transition Since EST is a step-wise prediction task, one entity can only undergo one action (transformation or location change) at each step. To properly evaluate EST on the sentence-level, there should be only one action per step for each entity.

(3) *Fallen rain or snow collects into surface water which will evaporate into water vapor again.*

In example 3, the entity *surface water* is assigned with two actions in one step, i.e., created from rain or snow and transformed into vapor by the annotators. A single timestep now contains two incompatible gold labels. Any model forced to choose one incurs a false negative on the other, capping maximum attainable performance.

Duplicate transition Sometimes, the same change is described across more than one step.

(4) *Step t: The light energy is used to convert carbon dioxide.*

...

Step t+2: The plant uses carbon dioxide in the air to produce glucose.

In example 4, step t and step $t+2$ together describe the creation of glucose. Annotating CREATE in both steps does not correctly represent the state changes of the involved entities. Models that avoid double-counting a one-off event are scored as under-predicting, while models that parrot the duplication are rewarded.

3.3 Violations of Complete State Accounting

Finally, ProPara often omits or mis-codes required state information.

Overgeneralized locations When annotating the location of a participant, the annotations sometimes are too general when there is a more specific mention of the location.

(5) *Blood returns to left side of your heart.*

In example 5, the location of *"blood"* is annotated as heart, which is technically correct but is not informative enough. So a model that predicts the finer span is marked wrong

Inconsistent state coding The same action does not always get the same annotation. For example, the event *"die"* is sometimes annotated as DESTROY and sometimes annotated as NONE since the annotators think the remains of that entity still exist.

Missing implicit transition When an action can only be inferred from the context or when the input/output is not explicitly stated, the corresponding state transition can be overlooked by the annotators. This happens most frequently when there is a transformation that involves multiple inputs where some inputs being explicitly mentioned while the

others not. The state changes of the entities that do not get mentioned explicitly are usually missing in the annotations.

(6) *Step t: A larvae matures inside of the egg.*

Step t+1: The caterpillar hatches from the egg.

In example 6, a transformation of larvae to caterpillar happens in step $t + 1$. However, since the action is not explicitly mentioned in the paragraph, the annotators miss to annotate the state of *"larvae"* as DESTROY. Thus a model that infers the disappearance of the larva is scored as incorrect, disincentivizing genuine causal reasoning

4 Re-annotating ProPara

To align ProPara with the CRTS constraints in §3, we perform a systematic re-annotation based on the Dense Paraphrasing (DP) technique (Ye et al., 2022; Tu et al., 2023). We treat DP as a truth-preserving textual enrichment strategy ϕ that maps a textual unit u (clause/sentence) to an enriched form $u^+ = \phi(u)$ in which otherwise implicit event roles, entity distinctions, and state transitions licensed by local lexical and discourse context are made explicit, with semantic fidelity $u^+ \Rightarrow u$. In this work, DP targets explicitness over economy by realizing missing arguments/roles, canonicalizing discourse referents, and expressing step-level state changes that are only implicit in u.

4.1 Re-annotation Actions

Resolving referent confusions via DP In the schema of ProPara-CRTS, we leverage DP to manually enrich entities with the same mention name so that their names become distinguishable while also reflecting the contexts. These enriched names establish a one-to-one mapping from the participant set P to the discourse-entity set E, ensuring that every participant has a single canonical referent and that no reference confusion can arise when models predict its state.

Whenever two lexically identical mentions denoted distinct discourse entities, the annotators should enrich these entities in DP style so that the names of these entities become distinct and can be easily linked to their corresponding entities. The names in the participant sets are subject to change per the names of entities they refer to. If necessary, the annotators will also add more participants to the participant set in one-to-many split situations like part-whole splits and conditional sentences.

In Figure 1, the *"tadpole"* in Step 5 should be re-annotated according to our new schema. It is re-annotated as *"new tadpole"* distinguishing it from the *"tadpole"* in Step 1. Also, *"new"* suggests that it is chronologically created in later steps.

Enforcing temporal atomicity Sentences that assign multiple transitions to one referent are divided into separate steps, or additional DP-distinguished referents are introduced so that each step contained exactly one action per entity. Duplicate transitions express across steps are merged into a single canonical event.

Completing state accounts For every referent at every step annotators record both an existence value and the most specific location span available. Missing implicit transitions (e.g. DESTROY of larva in the larva $\rightarrow$ caterpillar transformation) are inserted. Inconsistent action labels are also corrected.

Paragraph Issues Some paragraphs in ProPara are not a description of a process. One example describes how liver works with each sentence explaining a function of the liver. This kind of paragraphs are usually explanations where steps are not in order and few state or location changes are mentioned rather than process narratives. We believe these paragraphs do not qualify as process narratives and should not be included in the dataset. There are 36 paragraphs that fit in this category, and we exclude them from ProPara-CRTS.

4.2 Annotation Protocol and Qualiaty Control

Three trained graduate students with a background in Computational Linguistics annotate each paragraph in two passes. In Pass 1 the text is screened for non-process narratives; rejected paragraphs are removed. In Pass 2 manual error identification and CRTS corrections are applied, with DP edits logged and adjudicated when necessary. On a stratified sample of 100 paragraphs, we achieve a pairwise Cohen's κ of 0.72 for state labels and a pairwise F1 of 0.56 for location agreement.

4.3 Corpus Statistics

The re-annotated dataset, ProPara-CRTS, consists of 452 paragraphs and 13,417 state annotations. A total of 1,661 re-annotations were performed, encompassing updates to paragraphs, states, and par-

Group	Error type	Count	%
Referent Uniqueness	Name confusion	198	25.1
	Part–whole split	50	6.3
	Conditional branch	63	8.0
	Subtotal	**311**	**39.4**
Temporal Atomicity	Multiple transitions	40	5.1
	Duplicate transitions	45	5.7
	Subtotal	**85**	**10.8**
Complete State Accounting	Missing implicit transition	209	26.5
	Overgeneralized location	81	10.3
	Inconsistent state coding	67	8.5
	Subtotal	**357**	**45.2**
Paragraph Issues	Non-process paragraph	36	4.6
	Subtotal	**36**	**4.6**
Total		**789**	**100**

Table 1: Distribution of error types corrected during the Canonical Referent Tracking re-annotation of ProPara. Groups correspond to the three CRTS principles plus non-process paragraph issues.

ticipants, representing approximately 11% of the annotations in the original ProPara dataset. Most corrections in ProPara-CRTS address missing implicit changes, which accounts for 26.5% of all corrected errors. Name confusion and overgeneralizd location are the next two most frequent issues. The dataset maintains the same data splits as ProPara, with each partition corresponding to a subset of the original dataset partitions. Table 1 shows the statistics of corrected errors in ProPara-CRTS.

By integrating DP into the CRTS workflow, the new corpus removes referential drift without altering the underlying prose, thereby providing a sharply defined benchmark for measuring genuine model capacity in EST.

5 Experiments

In order to investigate the effectiveness of the new annotation, we evaluate different models against ProPara-CRTS and compare the results with those of the original ProPara dataset.

5.1 LLM Prompting

Considering the moderate modifications made to the original ProPara dataset, we believe that utilizing LLMs to perform inference on both test sets provides a valid basis for comparison.

We follow the experiment setups in MeeT (Singh et al., 2023) and frame EST into two subtasks: 1. A multi-choice problem where we ask the LLMs to select the state change of an entity from a fixed label set in step t. Similar to previous works (Zhang et al., 2021; Ma et al., 2022), we define six state types CREATE, NOT_CREATED, EXIST, MOVE, DESTROY, and WAS_DESTROYED. During evaluation, label NOT_CREATED, EXIST and WAS_DESTROYED will be mapped back to NONE. This enrichment of state label space helps the model differentiate the NONE types. 2. An extractive QA task that asks LLMs to extract the location of an entity in step t from the paragraph. Specifically, for each participant p, at each step t in the paragraph, we ask the LLMs two questions: 1) What is the state of p in step t? 2) Where is p located in step t? To preclude information leakage, the paragraph passed to the model is truncated at step t.

We compare four prompting strategies that differ only in the reasoning scaffold they present to the LLM while leaving the task formulation unchanged. The direct prompt elicits a terse answer and is decoded greedily with temperature 0. A Chain-of-Thought (CoT) variant adds the instruction "think step by step" and uses temperature 0.2 to encourage mild lexical diversity. The Self-Consistency setting draws eight independent CoT completions at temperature 0.7 and returns the state chosen by majority vote; when location spans disagree, the longest common subsequence is selected. Finally, a Few-Shot prompt supplies two worked examples drawn from the training split, each consisting of a short paragraph, the target entity, and the gold state and location pair.

We run the experiments using GPT-4o-mini and GPT-4o on the test sets of both ProPara and ProPara-CRTS. The detailed prompting queries to

Prompt	Dataset	Sent-level	Doc-level
Direct	ProPara	37.1	**59.6**
	ProPara-CRTS	**37.5**	58.8
CoT	ProPara	38.4	60.8
	ProPara-CRTS	**40.9**	**61.4**
Self-Cons.	ProPara	40.2	62.1
	ProPara-CRTS	**41.9**	**63.5**
Few-Shot	ProPara	39.0	61.3
	ProPara-CRTS	**40.0**	**62.2**

Table 2: Sentence-level and document-level F1 obtained by GPT-4o under four prompting scaffolds tested on ProPara and ProPara-CRTS test sets respectively.

Model	Dataset	Sent-level	Doc-level
GPT-4o-mini	ProPara	**24.8**	55.0
	ProPara-CRTS	23.2	**55.6**
GPT-4o	ProPara	40.2	62.1
	ProPara-CRTS	**41.9**	**63.5**
FT Llama 3.1	ProPara	59.5	64.9
	ProPara-CRTS	**63.6**	**65.9**
MeeT	ProPara	54.9	69.4
	ProPara-CRTS	**61.9**	**70.8**
CGLI	ProPara	65.4	72.4
	ProPara-CRTS	**69.5**	**75.1**

Table 3: Sentence-level and document-level evaluation results (F1) of models tested on ProPara and ProPara-CRTS test sets respectively.

the LLMs are shown in Appendix A.2.

5.2 LLM Fine-tuning

We also try fine-tuning LLMs for the EST task on the ProPara datasets. Due to the limitation of computing resources, we decide to fine-tune a smaller model `Llama 3.1 8B`[2] twice: once on ProPara and once on ProPara-CRTS. We fine-tune using the training sets of each dataset and evaluate their performance on the test sets of each dataset.

To fine-tune Llama 3.1, we make several adjustments to improve efficiency. We use 4-bit quantized models and LoRA (Hu et al., 2021) layers on all seven target modules available on Llama models. In addition, we use the `unsloth` method for gradient checkpointing. We use the CoT prompt during fine-tuning and Self-Consistency prompt for inference. Full fine-tuning hyperparameter set is reported in Appendix A.1.

5.3 Supervised Learning Models

We evaluate ProPara-CRTS with two supervised learning models MeeT (Singh et al., 2023) and CGLI (Ma et al., 2022), which are the top 2 models on the ProPara leaderboard[3]. MeeT formulates the EST into two subtasks: state prediction and location prediction. Then it fine-tunes the T5 model on both tasks. CGLI uses RoBERTa and leverages a decoding strategy that considers the context of each step on both local and global levels. We reuse the hyperparameters and settings reported by the authors to train the two models on ProPara-CRTS.

6 Results

Table 2 reports GPT-4o's performance under four prompting scaffolds, while Table 3 places those results alongside parameter-efficient fine-tuning and fully-supervised baselines.

Evaluation scheme Following the original ProPara setup, models are judged by their ability to answer three QA categories derived from an entity–step grid: (Cat1) whether an entity is created/destroyed/moved, (Cat2) at which step the change occurs, and (Cat3) where it occurs (Dalvi et al., 2018). In our reporting, *sentence-level* scores evaluate these predictions at the granularity of entity–step pairs, aggregated as macro/micro F_1 over Cat1–3. Complementarily, *document-level* scores treat all answers for a paragraph as a set of predicted tuples (entity, change-type, step, locations) and compute precision/recall/F_1 against the gold set, emphasizing global consistency across the whole process (Tandon et al., 2018).

Effect of canonical reference Across all settings ProPara-CRTS yields higher F1 scores than the original annotation except for GPT-4o-mini at sentence-level. For GPT-4o the absolute gain ranges from +0.4 points with the direct prompt to +1.7 points with self-consistency. The trend is even clearer for trained models: Llama-3-8B fine-tuned on CRTS improves by +4.1 points at the sentence level and by +1.0 points at the document level; CGLI and MeeT register gains of +4.1 and +7.0 points respectively. These consistent improvements confirm that enforcing a canonical, one-to-one referent mapping removes annotation noise that previously capped model scores.

Impact of reasoning scaffolds Moving from the direct question to Chain-of-Thought (CoT) increases GPT-4o's sentence-level score from 37.1 to 38.4 on ProPara and from 37.5 to 40.9 on CRTS. Adding self-consistency sampling delivers a further

[2] `unsloth/Meta-Llama-3.1-8B-Instruct-bnb-4bit`
[3] https://leaderboard.allenai.org/propara/submissions/public

gain, reaching 41.9 / 63.5 on CRTS—the best zero-shot result in our study. Few-shot prompting also helps, though its improvement is slightly smaller than that of self-consistency. The pattern suggests that canonical referent tracking particularly benefits prompts that require multi-step inference: once referential ambiguity is removed, the model's explicit reasoning is more likely to be correct and internally consistent.

Prompting versus training Even with the strongest scaffold, GPT-4o remains 22 points below the best supervised model (CGLI) at the sentence level and 12 points below at the document level. Fine-tuning a relatively small 8-billion-parameter Llama eliminates more than half of that gap, surpassing the older supervised systems MeeT and CGLI on the original corpus and approaching them on CRTS. These results indicate that canonical referent tracking narrows but does not erase the difference between prompt-only and parameter-updated approaches; models still gain substantially from task-specific training.

Canonical referent tracking lifts every method we tested, but the magnitude of the lift is modulated by the model's ability to exploit richer reasoning scaffolds or supervised updates. Prompt-engineering alone can reach the mid-40s F1 at the sentence level, yet fine-tuning remains essential for closing the gap to state-of-the-art supervised systems. We show the full results in Appendix A.3.

	Sent-level		Doc-level	
Model	ProPara	ProPara-CRTS	ProPara	ProPara-CRTS
FT Llama 3.1	61.9	**63.6**	65.0	**65.9**
MeeT	59.9	**61.9**	70.2	**70.8**
CGLI	67.2	**69.5**	71.1	**75.1**

Table 4: Cross-dataset evaluation results (F1) of models trained on the training sets of ProPara and ProPara-CRTS and tested against ProPara-CRTS test set.

	Sent-level		Doc-level	
Model	ProPara	ProPara-CRTS	ProPara	ProPara-CRTS
FT Llama 3.1	62.3	**64.6**	65.4	**67.5**
MeeT	60.8	**62.8**	70.8	**71.3**
CGLI	68.3	**71.2**	73.3	**76.5**

Table 5: Cross-dataset evaluation results (F1) of models trained on the training sets of ProPara and ProPara-CRTS and tested against the shared slice of ProPara and ProPara-CRTS test sets.

7 Analysis

To further investigate the effectiveness of the re-annotation, we look into the prediction differences in ProPara and ProPara-CRTS and see if the models trained on ProPara-CRTS understand process narratives better.

7.1 Cross-Dataset Evaluation

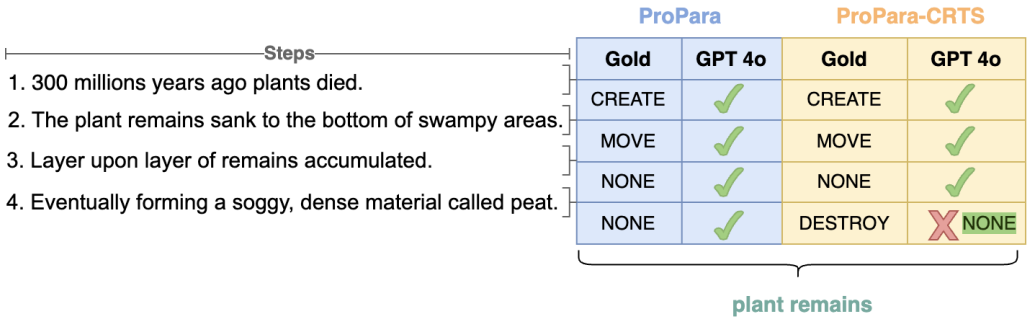
We perform a cross-dataset evaluation by training identical models on the training sets of ProPara and ProPara-CRTS, and then assessing their performance on ProPara-CRTS test set. Notably, models trained on ProPara all exhibit a decline in accuracy compared to those trained and tested exclusively on ProPara-CRTS, underscoring the critical value of high-quality training data and the advantages offered by ProPara-CRTS. The main findings are presented in Table 4.

We further evaluate models on the shared slice of both ProPara and ProPara-CRTS test sets, where no additional CRTS-only annotations are available. The intersection test set contains 42 paragraphs out of 52 of the original ProPara test set. The results are shown in Table 5. Training on ProPara-CRTS improves performance even when evaluation is restricted to the intersection of both test sets. Gains are consistent across all architectures. Because the test slice excludes CRTS-specific enrichment, these improvements indicate better generalization from cleaner training signals rather than artifacts of a richer label space.

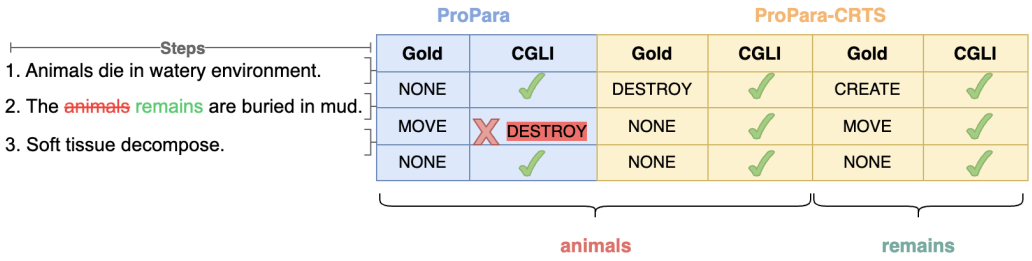
Comprehensive results of both experiments are reported in Appendix A.3.

7.2 Qualitative Analysis

Qualitatively, we compare model predictions obtained from training on both datasets and observe that models trained on ProPara-CRTS effectively mitigate the shortcomings inherent in the original annotations. As illustrated in Figure 2a, we show the same paragraph in ProPara and ProPara-CRTS with different state annotations. In Step 4, the entity *plant remains* undergoes a transformation and forms into peat. The annotation in ProPara misses this transition because the input entity, *plant remains*, is not explicitly stated. The annotation in ProPara-CRTS corrects it. The prediction from GPT-4o on Step 4 is NONE as it fails to identify this implicit action as well. This mistake is also regarded as correct when evaluating against ProPara.



(a) State predictions of entity *plant remains* by GPT-4o.



(b) State predictions of entities *animals* and *remains* by CGLI.

Figure 2: State predictions on the same paragraphs or their re-annotated counterparts in ProPara and ProPara-CRTS. The check denotes that the predictions match the gold. The cross denotes a mismatch with the gold. Green background of the prediction means it is factually correct, red otherwise.

This shows that if the gold annotations are problematic, the evaluation results can be misleading.

Figure 2b demonstrates an example where the re-annotation help CGLI better predict the states. In the example, CGLI is asked to track the state of participant *"animals"*. However, CGLI fails to identify that *die* is an action of DESTROY, and predicts that there is no state change for *"animals"* in Step 1. We suspect that this is because there is another mention of *"animals"* in Step 2 so CGLI assumes that the animals are still alive in Step 1. This is a mistake caused by reference confusion where the *"animals"* in Step 1 refers to living animals while the mention in Step 2 refers to animal remains, which should be differentiated in EST. Hence, CGLI is actually tracking the states of the wrong entity. By decontextualizing the *"animals"* in Step 2, we distinguish the two entities which share the same name. And the example shows that CGLI is able to predict the states of both participants correctly. This indicates that the canonical referent tracking schema help model to better comprehend process narratives.

8 Conclusion

We have presented ProPara-CRTS, a rigorously re-annotated version of the ProPara corpus that replaces the original, ambiguity-prone schema with a Canonical Referent Tracking Schema. CRTS enforces one-to-one mention–referent mapping, stepwise atomicity, and exhaustive state accounting. DP supplies the minimal lexical edits needed to make colliding mentions distinguishable while preserving the original prose. During re-annotation we also corrected recurrent paragraph and state-label errors, yielding 452 paragraphs with 13,417 noise-free state triples. Experiments spanning from LLM prompting, LLM fine-tuning and supervised models show consistent gains on CRTS. The results confirm three claims: (i) referential canonicalization removes label noise that previously suppressed scores; (ii) prompts that elicit multi-step reasoning profit most from the cleaner supervision; and (iii) despite these gains, EST remains challenging—supervised models still outperform purely prompted LLMs, underscoring the importance of dedicated training data. We release ProPara-CRTS, annotation guidelines, and validation scripts to facilitate future work on robust, semantics-aware evaluation of EST in natural-language process narratives.

9 Limitation

Due to resource constraints, only a subset of 100 paragraphs from ProPara underwent dual re-annotation, while the remaining paragraphs were subjected to single re-annotation. Consequently, the inter-annotator agreement was calculated solely based on this limited sample. Furthermore, for each paragraph, both the re-annotation of the paragraph text and the state labels were conducted by the same annotator, which could introduce potential bias into the annotations. Despite the involvement of three specially-trained annotators, the possibility of unintentional errors or subjective judgments remains.

References

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D. Manning. 2014. Modeling biological processes for reading comprehension. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1499–1510, Doha, Qatar. Association for Computational Linguistics.

Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2017. Simulating action dynamics with neural process networks. *arXiv preprint arXiv:1711.05313*.

Bhavana Dalvi, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. 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 1595–1604, New Orleans, Louisiana. Association for Computational Linguistics.

Yanai Elazar, Victoria Basmov, Yoav Goldberg, and Reut Tsarfaty. 2022. Text-based NP enrichment. *Transactions of the Association for Computational Linguistics*, 10:764–784.

Biaoyan Fang, Timothy Baldwin, and Karin Verspoor. 2022. What does it take to bake a cake? the RecipeRef corpus and anaphora resolution in procedural text. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3481–3495, Dublin, Ireland. Association for Computational Linguistics.

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.

Jinning Li, Shubhanshu Mishra, Ahmed El-Kishky, Sneha Mehta, and Vivek Kulkarni. 2022. NTULM: Enriching social media text representations with non-textual units. In *Proceedings of the Eighth Workshop on Noisy User-generated Text (W-NUT 2022)*, pages 69–82, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Reginald Long, Panupong Pasupat, and Percy Liang. 2016. Simpler context-dependent logical forms via model projections. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1456–1465, Berlin, Germany. Association for Computational Linguistics.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Eric Nyberg, and Alessandro Oltramari. 2022. Coalescing global and local information for procedural text understanding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1534–1545, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Pierre André Ménard and Antoine Mougeot. 2019. Turning silver into gold: error-focused corpus re-annotation with active learning. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 758–767, Varna, Bulgaria. INCOMA Ltd.

Bhavana Dalvi Mishra, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. *arXiv preprint arXiv:1805.06975*.

Saed Rezayi, Handong Zhao, Sungchul Kim, Ryan Rossi, Nedim Lipka, and Sheng Li. 2021. Edge: Enriching knowledge graph embeddings with external text. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2767–2776, Online. Association for Computational Linguistics.

Kyeongmin Rim, Jingxuan Tu, Bingyang Ye, Marc Verhagen, Eben Holderness, and James Pustejovsky. 2023. The coreference under transformation labeling dataset: Entity tracking in procedural texts using event models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12448–12460, Toronto, Canada. Association for Computational Linguistics.

Janvijay Singh, Fan Bai, and Zhen Wang. 2023. Entity tracking via effective use of multi-task learning model and mention-guided decoding. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1255–1263, Dubrovnik, Croatia. Association for Computational Linguistics.

Shane Storks, Qiaozi Gao, Yichi Zhang, and Joyce Chai. 2021. Tiered reasoning for intuitive physics: Toward

verifiable commonsense language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4902–4918, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Niket Tandon, Bhavana Dalvi, Joel Grus, Wen-tau Yih, Antoine Bosselut, and Peter Clark. 2018. Reasoning about actions and state changes by injecting commonsense knowledge. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 57–66, Brussels, Belgium. Association for Computational Linguistics.

Niket Tandon, Bhavana Dalvi, Keisuke Sakaguchi, Peter Clark, and Antoine Bosselut. 2019. WIQA: A dataset for "what if..." reasoning over procedural text. 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 6076–6085, Hong Kong, China. Association for Computational Linguistics.

Niket Tandon, Keisuke Sakaguchi, Bhavana Dalvi, Dheeraj Rajagopal, Peter Clark, Michal Guerquin, Kyle Richardson, and Eduard Hovy. 2020. A dataset for tracking entities in open domain procedural text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6408–6417, Online. Association for Computational Linguistics.

Jingxuan Tu, Kyeongmin Rim, Eben Holderness, Bingyang Ye, and James Pustejovsky. 2023. Dense paraphrasing for textual enrichment. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 39–49, Nancy, France. Association for Computational Linguistics.

Jingxuan Tu, Keer Xu, Liulu Yue, Bingyang Ye, Kyeongmin Rim, and James Pustejovsky. 2024. Linguistically conditioned semantic textual similarity. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1161–1172, Bangkok, Thailand. Association for Computational Linguistics.

Weiqi Wang and Yangqiu Song. 2024. Mars: Benchmarking the metaphysical reasoning abilities of language models with a multi-task evaluation dataset.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Xueqing Wu, Sha Li, and Heng Ji. 2023. OpenPI-C: A better benchmark and stronger baseline for open-vocabulary state tracking. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7213–7222, Toronto, Canada. Association for Computational Linguistics.

Bingyang Ye, Jingxuan Tu, Elisabetta Jezek, and James Pustejovsky. 2022. Interpreting logical metonymy through dense paraphrasing. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.

Li Zhang, Hainiu Xu, Abhinav Kommula, Chris Callison-Burch, and Niket Tandon. 2024. OpenPI2.0: An improved dataset for entity tracking in texts. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 166–178, St. Julian's, Malta. Association for Computational Linguistics.

Zhihan Zhang, Xiubo Geng, Tao Qin, Yunfang Wu, and Daxin Jiang. 2021. Knowledge-aware procedural text understanding with multi-stage training. In *Proceedings of the Web Conference 2021*, pages 3512–3523.

rank	16
lora_alpha	16
lora_dropout	0
target_modules	q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj
max_seq_length	2048
use_gradient_checkpointing	unsloth
per_device_train_batch_size	2
gradient_accumulation_steps	4
warmup_steps	5
num_train_epochs	1
learning_rate	2e-4
optim	adamw_8bit
weight_decay	0.01
lr_scheduler_type	linear

Table A1: Hyperparameters for Unsloth fine-tuning.

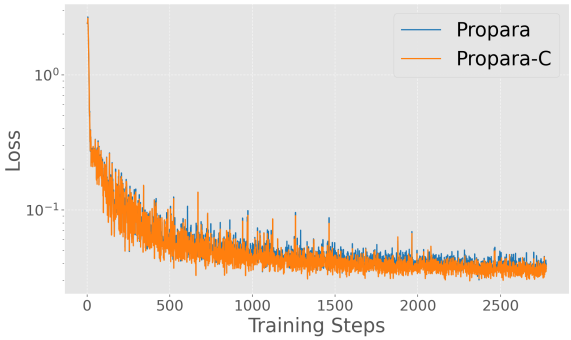


Figure A1: Fine-tuning loss on ProPara and ProPara-CRTS using Llama 3.1.

A Appendix

A.1 LLMs Fine-tuning

We report the hyperparameters for fine-tuning in Table A1. During fine-tuning Llama 3.1 on both datasets, the loss quickly decreases and then stabilizes during the first epoch of training. Therefore, we stop fine-tuning after one epoch. As shown in Figure A1, the loss functions for both datasets are nearly identical. This is expected as the task itself is not changing. The average loss for ProPara is 0.06618. For ProPara-CRTS, it is 0.06341, only a 4% difference.

While it is not a difficult task for humans, LLMs struggle to be competitive on the EST task. We believe this is due to the ambiguity associated with the task. Even with a limited set of responses, annotators will interpret these labels differently. The sharp initial decrease in loss is where the model learns the expected format of the answers. Very soon after starting training, the model produces correctly formatted responses, but they are less accurate than those collected after fine-tuning has concluded.

A.2 Prompts

Figure A2 illustrates the direct prompts we feed to LLMs for inference. Figure A3 demonstrates the CoT prompts we use for LLMs inference and fine-tuning. We also use the same prompt for self-consistency setting. Figure A4 shows the few-shot prompts for LLMs inference.

A.3 Results

We report the full sentence-level and document-level evaluation results of models on ProPara and ProPara-CRTS in Table A2. We report the cross-dataset evaluation results in Table A3 where EST models are trained on ProPara and ProPara-CRTS training sets respectively and tested on ProPara-CRTS test set. We report the cross-dataset evaluation results in Table A4 where EST models are trained on ProPara and ProPara-CRTS training sets respectively and tested on the shared slice of ProPara and ProPara-CRTS test sets.

Model / Train set		Sentence-level					Document-level		
Model	Train	Cat1	Cat2	Cat3	Macro	Micro	P	R	F1
GPT-4o-mini	ProPara	59.3	9.5	0.6	23.4	**24.8**	70.2	44.2	55.0
	ProPara-CRTS	53.7	10.8	0.6	21.7	23.2	68.2	41.9	**55.6**
GPT-4o	ProPara	70.8	36.4	11.5	39.6	40.2	63.1	61.2	62.1
	ProPara-CRTS	67.7	36.5	13.8	39.3	**41.9**	62.3	53.4	**63.5**
FT Llama 3.1	ProPara	78.1	55.2	45.6	59.6	59.5	66.4	63.5	64.9
	ProPara-CRTS	79.9	61.6	50.2	63.9	**63.6**	64.9	66.9	**65.9**
MeeT	ProPara	77.0	50.8	37.8	55.1	54.9	79.0	61.9	69.4
	ProPara-CRTS	81.1	62.6	43.9	62.5	**61.9**	78.5	64.5	**70.8**
CGLI	ProPara	81.1	61.7	53.8	65.5	65.4	74.9	70.0	72.4
	ProPara-CRTS	83.9	70.5	55.6	70.0	**69.5**	80.3	70.6	**75.1**

Table A2: Sentence-level and document-level evaluation results of models on ProPara and ProPara-CRTS.

Model / Train set		Sentence-level					Document-level		
Model	Train	Cat1	Cat2	Cat3	Macro	Micro	P	R	F1
FT Llama 3.1	ProPara	77.2	58.0	50.8	62.0	61.9	68.1	62.2	65.0
	ProPara-CRTS	79.9	61.6	50.2	63.9	**63.6**	64.9	66.9	**65.9**
MeeT	ProPara	78.6	55.6	46.0	60.0	59.9	81.3	61.7	70.2
	ProPara-CRTS	81.1	62.6	43.9	62.5	**61.9**	78.5	64.5	**70.8**
CGLI	ProPara	81.6	65.5	55.3	67.5	67.2	75.7	67.0	71.1
	ProPara-CRTS	83.9	70.5	55.6	70.0	**69.5**	80.3	70.6	**75.1**

Table A3: Cross-dataset evaluation results of the setting where EST models are trained on ProPara and ProPara-CRTS respectively and tested on ProPara-CRTS.

Model / Train set		Sentence-level					Document-level		
Model	Train	Cat1	Cat2	Cat3	Macro	Micro	P	R	F1
FT Llama 3.1	ProPara	78.8	58.4	50.6	62.6	62.3	67.8	63.1	65.4
	ProPara-CRTS	78.5	61.5	52.3	64.1	**64.6**	69.2	65.9	**67.5**
MeeT	ProPara	79.1	56.2	46.3	60.6	60.8	81.0	62.9	70.8
	ProPara-CRTS	81.3	63.4	44.7	63.1	**62.8**	78.2	65.4	**71.3**
CGLI	ProPara	82.3	68.0	55.3	68.5	68.3	77.8	69.2	73.3
	ProPara-CRTS	84.4	72.3	57.9	71.5	**71.2**	79.2	73.9	**76.5**

Table A4: Cross-dataset evaluation results of the setting where EST models are trained on ProPara and ProPara-CRTS respectively and tested on the shared slice of ProPara and ProPara-CRTS test sets.

System Prompt

You are an intelligent assistance that can track the state of an given entity from a document. The following is the document.

Document:
`[1] A plant or animal dies in a watery environment.`
`[2] The remain is buried in mud and silt.`

State Tracking Instruction

Based on the document, answer the question: What is the state of `<plant; animal>` exactly after step [2]?
- Create — entity absent before step [2] but present after
- Not_Created — entity has not been created yet after step [2]
- Move — entity present before step [2] and after, but in a different location
- Destroy — entity present before step [2] but absent after
- Was_Destroyed — entity had already been destroyed before step [2]
- Exist — entity present before step [2] and after, in the same location

You should only answer with Create, Move, Destroy, Not_created, Was_destroyed and Exist without saying anything else.

Response

Move; gold: Exist

Location Tracking Instruction

Based on the document, answer the question: Where is `<plant; animal>` located exactly after step [2]?
If you cannot tell the location of `<plant; animal>` after [2], answer with 'unknown'. If `<plant; animal>` is either not created yet or was destroyed before [2], answer with 'not_exist'. The extracted text should be exactly the same with that in the document. Only respond with the extracted text, do not say any other words or explain.

Response

in mud and silt; gold: not_exist

Figure A2: Direct prompt for LLMs inference.

System Prompt

You are an intelligent assistance that can track the state of an given entity from a document. The following is the document.

Document:
`[1] A plant or animal dies in a watery environment.`
`[2] The remain is buried in mud and silt.\`

State Tracking Instruction

Based on the document, answer the question: What is the state of `<plant; animal>` exactly after step [2]?
- Create — entity absent before step [2] but present after
- Not_Created — entity has not been created yet after step [2]
- Move — entity present before step [2] and after, but in a different location
- Destroy — entity present before step [2] but absent after
- Was_Destroyed — entity had already been destroyed before step [2]
- Exist — entity present before step [2] and after, in the same location

You should only answer with Create, Move, Destroy, Not_created, Was_destroyed and Exist without saying anything else.

Think step by step about the entity's existence and location from step 1 through [2].

Reasoning:
(Your chain-of-thought here.)

Answer:
State = <LABEL>
Location = <SPAN | unknown | not_exist>

Response

Move; gold: Exist

Location Tracking Instruction

Based on the document, answer the question: Where is `<plant; animal>` located exactly after step [2]?
If you cannot tell the location of `<plant; animal>` after [2], answer with 'unknown'. If `<plant; animal>` is either not created yet or was destroyed before [2], answer with 'not_exist'. The extracted text should be exactly the same with that in the document. Only respond with the extracted text, do not say any other words or explain.

Think step by step about the entity's existence and location from step 1 through [2].

Reasoning:
(Your chain-of-thought here.)

Answer:
State = <LABEL>
Location = <SPAN | unknown | not_exist>

Response

in mud and silt; gold: not_exist

Figure A3: CoT Prompt for LLMs inference and fine-tuning.

System Prompt

You are an intelligent assistance that can track the state of an given entity from a document.
You are first given two examples.

Example 1
<Example1>

Example 2
<Example2>

The following is the target document.
Document:
`[1] A plant or animal dies in a watery environment.`
`[2] The remain is buried in mud and silt.`

State Tracking Instruction

Based on the document, answer the question: What is the state of `<plant; animal>` exactly after step [2]?
- Create — entity absent before step [2] but present after
- Not_Created — entity has not been created yet after step [2]
- Move — entity present before step [2] and after, but in a different location
- Destroy — entity present before step [2] but absent after
- Was_Destroyed — entity had already been destroyed before step [2]
- Exist — entity present before step [2] and after, in the same location

You should only answer with Create, Move, Destroy, Not_created, Was_destroyed and Exist without saying anything else.

Response

Move; gold: Exist

Location Tracking Instruction

Based on the document, answer the question: Where is `<plant; animal>` located exactly after step [2]?
If you cannot tell the location of `<plant; animal>` after [2], answer with 'unknown'. If `<plant; animal>` is either not created yet or was destroyed before [2], answer with 'not_exist'. The extracted text should be exactly the same with that in the document. Only respond with the extracted text, do not say any other words or explain.

Response

in mud and silt; gold: not_exist

Figure A4: Few-shot prompt for LLMs inference.

The Difficult Case of Intended and Perceived Sarcasm: a Challenge for Humans and Large Language Models

Hyewon Jang[1,3] **& Diego Frassinelli**[2]
[1]Department of Linguistics, University of Konstanz
[2]MaiNLP, Center for Information and Language Processing, LMU Munich
[3]Dept. of Philosophy, Linguistics and Theory of Science, University of Gothenburg
hyewon.jang@gu.se, frassinelli@cis.lmu.de

Abstract

We examine the cases of failed communication in sarcasm, defined as 'the discrepancy between what speakers and observers perceive as sarcasm'. We identify factors that are associated with such failures, and how those difficult instances affect the detection performance of encoder-only and decoder-only generative models. We find that speakers' incongruity between their felt annoyance and sarcasm in their utterance is highly correlated with sarcasm that fails to be communicated to human observers. This factor also relates to the drop of classification performance of large language models (LLMs). Additionally, disagreement among multiple observers about sarcasm is correlated with poorer performance of LLMs. Finally, we find that generative models produce better results with ground-truth labels from speakers than from observers, in contrast to encoder-only models, which suggests a general tendency by generative models to identify with speakers' perspective by default.

1 Introduction

An utterance that is intended to be sarcastic by the speaker is sometimes not understood as such by the listener or external observers, or vice versa (Fox Tree et al., 2020). Consider the example below.

> About two years ago, Steve spent half a year in Japan, where he learned a lot about Japanese food culture. Ever since then, whenever Steve and John eat something together, Steve says some version of, "you know, in Japan, people do it this way." And John says, "that's cool to hear!"

In this situation, if an external observer thinks that John is being sarcastic, but John intended to be literal, there is a discrepancy between intended and perceived sarcasm.[1] This type of communication failure can occur in numerous communicative scenarios, especially those requiring layers of inferences, which are common features of sarcasm (Bryant, 2023). Discussing the divergence between intended and perceived sarcasm is not new. Prior work in psycholinguistics has widely discussed the differences in what speakers intend with sarcastic utterances and how listeners or observers interpret them (Pexman and Olineck, 2002). NLP tasks and datasets are also affected by such discrepancies. For instance, Oprea and Magdy (2020) demonstrated that there are many instances for which external annotators provide different sarcasm judgments from the producers of the utterances. Sarcasm detection by language models, especially BERT-like models, also show different classification performances according to ground-truth labels based on self-evaluation versus external evaluation (Abu Farha et al., 2022; Jang and Frassinelli, 2024; Oprea and Magdy, 2019; Plepi and Flek, 2021). Since the capacity of (large) language models has increased exponentially over the years with the advent of generative models, which are often placed in direct conversations with human users, it has become an important question to ask how language models navigate different perspectives in communication involving sarcasm.

Although it is evident that sarcasm judgment is contingent on the different perspectives of speakers and observers (Oprea and Magdy, 2020), there is a lack of systematic investigation on what factors contribute to the general difficulty of the task for LLMs as well as for human observers. One of the numerous keys to identifying the source of such discrepancy between speakers and observers is to think about why sarcasm is used in the first place.

[1]In this work, the discrepancy we address is between speaker and observer rather than speaker and listener. An observer is a non-participant of a conversation who evaluates the interaction from an external perspective. Though this is less natural for real communication, it is more relevant for computational linguistics, as data are often created with evaluation by external observers.

Proceedings of the 16th International Conference on Computational Semantics, pages 279–291
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

Sarcasm is used to convey specific communicative intentions, such as to mock the addressee (Gibbs, 2000), which in turn is motivated by speaker's emotion in a given communicative situation (Jang et al., 2023). The strong link between emotion and sarcasm has long been identified and discussed in numerous previous studies (e.g., Filik, 2023; Jang et al., 2023; Veale, 2023). As such, we focus on the close connection between sarcasm and emotion to examine the discrepancy between speakers and external observers in the use of sarcasm. Specifically, we focus on *annoyance*, an emotion shown to strongly influence sarcasm production and identification (Jang et al., 2023). Though annoyance is not the only reason why a speaker chooses to use sarcasm, we focus on annoyance in this work based on Jang et al. (2023), who report that a strong connection is observed between speaker's annoyance and the level of sarcasm in their utterance, and that external observers are also able to capture this connection. The availability of such information in the dataset described in Section 3.1 also motivates such research design.[2] We demonstrate which factors are associated with the divergence of sarcasm judgment between speakers and observers and how this affects (L)LM performance on sarcasm detection.

2 Related work

2.1 Intended vs. perceived sarcasm

Numerous previous studies exist on sarcasm detection, but very few of them address the perspective divergence between speakers and external observers (Dadu and Pant, 2020; Khodak et al., 2018; Kumar and Anand, 2020; Misra and Arora, 2023). In fact, an absolute majority of sarcasm datasets contain labels annotated by third-party annotators (Castro et al., 2019; Khodak et al., 2018; Oraby et al., 2016), or a combination of self-labels and third-party labels (Khodak et al., 2018; Van Hee et al., 2018). Some datasets provide only author labels without third-party labels (Oprea and Magdy, 2020). Only a small body of work addresses the difference between intended and perceived sarcasm (Jang et al., 2023; Jang and Frassinelli, 2024; Oprea and Magdy, 2020; Plepi and Flek, 2021; Shmueli et al., 2020). They report that there is a noticeable difference in LM performance depending on the source of ground-truth labels. But the discussion of which factors may contribute to such difference, or how the difference can be used to evaluate LLM performance has not been extensively addressed in the literature.

2.2 The connection between sarcasm and emotion

Previous work has identified numerous reasons for which human communicators use sarcasm. Sarcasm can be used to express an attitude (Colston, 2023), to cause certain emotional reactions in the listener (Filik, 2023), or to achieve specific communicative goals such as to be humorous (Gibbs, 2000), appear emotionally controlled (Dews et al., 1995), or mock the addressee (Pexman and Olineck, 2002). These communicative functions are often motivated by the emotion in reaction to an experience (Jang et al., 2023). Sarcasm as such is strongly related to emotions, whether sarcasm serves as the trigger for emotional reactions or is itself triggered by them.

3 Method

3.1 Data

We used the publicly available Conversational Sarcasm Corpus (CSC; Jang and Frassinelli, 2024)[3] to analyze misaligned cases between intended and perceived sarcasm. CSC provides a good opportunity to examine divergences in sarcasm judgment because it provides evaluations of two concepts (sarcasm & emotion) reported by both speakers and multiple external observers (4-6 per speaker). Specifically, it contains contexts and utterances (N = 7,036), ratings for *sarcasm* and *annoyance felt by the speaker* that are judged by two parties (speaker & observers). The original ratings provided in the dataset are text-coded as *1 (not at all) - 2 (mostly not) - 3 (not so much) - 4 (somewhat) - 5 (mostly) - 6 (completely)*, which makes both numerical manipulation and binarization possible.

3.2 Hypotheses

We identified two potential sources of gap for which observers reach a different judgment about sarcasm than the speakers:

[2] We further tested the validity of annoyance as a relevant emotion to sarcasm in a separate preliminary experiment using an emotion classification model (https://huggingface.co/bsingh/roberta_goEmotion) fine-tuned on the GoEmotions dataset (Demszky et al., 2020). When using this model, the logits of the top 20% of most important emotions (out of 28 categories) for sarcastic utterances from CSC were *annoyance, admiration, amusement, approval*, and *curiosity*.

[3] https://github.com/CoPsyN/CSC

Type	Text	Sarc(S)	Sarc(O)	Ann(S)	Ann(O)
H1: Speaker's annoyance-sarcasm incongruity	Context: You got a date this evening. When you tell Steve you got a date, he asks, "oooh, what's the plan?" Response: ***We're going for Malaysian and then a gig.***	6	1	2	1
H2: Speaker-observer annoyance misalignment	Context: About two years ago, Steve spent half a year in Japan, where he learned a lot about Japanese food culture. Ever since then, whenever you eat something together, Steve says some version of, "you know, in Japan, people do it this way." Response: ***That's cool to hear!***	1	6	1	6

Table 1: Examples of *speaker's annoyance-sarcasm incongruity* (boxed) and *speaker-observer annoyance misalignment* (underlined) associated with sarcasm failure (6 vs. 1). Sarc=Sarcasm ratings, Ann=Annoyance ratings, S=Speaker, O=Observer.

- **H1: speakers' annoyance-sarcasm incongruity**: An incongruity between a speaker's annoyance and the level of sarcasm in the output utterance causes misalignment between self-rated and other-rated sarcasm.
- **H2: observers' failure to detect annoyance**: A failure by observers to identify the annoyance a speaker felt in a given situation causes the misaligned sarcasm judgment between speakers and observers.

Table 1 shows the cases from the data that exemplify either **H1** or **H2**. In both cases, sarcasm has failed to be communicated, since the ratings given by the speakers and observers are substantially different (6 vs. 1). However, in each case, different factors stand out as being associated with the failure. In **H1**, we hypothesize that the gap between the sarcasm rating and annoyance rating given by the speaker (6 vs. 2) may be associated with the failure of communicating sarcasm (*speaker's annoyance-sarcasm incongruity*). In **H2**, we hypothesize that the discrepancy between the annoyance ratings given by the speaker and observer (1 vs. 6) is linked to the failure of communication (*observers' failure to identify speaker's annoyance*).

4 Experiment 1: Sarcasm detection by human observers

The first experiment inspected the factors related to sarcasm communication failure between human speakers and observers, by testing two hypotheses described in Section 3.2.

4.1 Quantifying variables

Sarcasm alignment: We quantified the alignment between the sarcasm scores given by a speaker and multiple observers using the inverse of normalized mean absolute error (MAE).[4] We define alignment as:

$$1 - \frac{1}{n}\sum_{i=1}^{n}|y - \hat{y}|$$

where y is the rating by the speaker, $\hat{y}$ is the rating from an observer for the same instance, and n is the total number of speaker-observer pairs considered. Values closer to 1 indicate stronger agreement. Sarcasm alignment score is the dependent variable in our following statistical analysis.

Annoyance alignment: The alignment between annoyance scores given by a speaker and multiple observers is computed using the same formula as above. This measure is one of the main predictors in our statistical analyses.

Speaker congruity: To quantify the congruity between sarcasm and annoyance expressed by the speaker we assigned a value of 1 (*congruous*) if the speaker rated sarcasm and annoyance levels as both negative (*1-not at all, 2-mostly not, 3-not so much*) or positive (*4-somewhat, 5-mostly, 6-completely*). If the speaker gave a negative rating to sarcasm (1, 2, 3) but a positive rating to annoyance (4, 5, 6), and vice versa, we assigned a value of 0 (*incongruous*). Together with annoyance alignment, this is the second main predictor in our statistical analysis.

	SP	OB1	OB2	OB3	OB4	OB5	OB6	Avg	**Alignment**
Ex.1	4	5	4	5	4	4	1	3.83	0.86
Ex.2	4	5	6	4	3	2	3	3.83	0.81

Table 2: Examples with the same average score (Avg) but with different alignment scores between speaker and observers.

[4]Though a conventional measure for quantifying errors is the mean squared error (MSE), the mean absolute error (MAE) aligns with the purpose of our task better, because the MAE does not penalize outliers among observers as harshly as the MSE. A single outlier is not much of a communication failure as long as the majority of the observers make judgments similar to the speaker's original intention.

The two examples in Table 2 have the same average score, but in Example 1, most observers agreed with the speaker except for one major outlier, while Example 2 shows less alignment overall between the observers and the speaker. Therefore, Example 1 gets a higher alignment score of 0.86 and Example 2 gets a lower score of 0.81.

We tested our hypothesis using a linear mixed-effects model (Barr et al., 2013) that predicted **sarcasm alignment** given the **annoyance alignment** in interaction with **speaker's annoyance-sarcasm congruity**, with by-item and by-participant random intercepts.

4.2 Results

The speaker's annoyance-sarcasm **congruity** showed a statistically significant positive effect on speaker-observer **sarcasm alignment** ($\beta = 0.15, p < 0.001$). Importantly, we found a strong positive interaction effect between the two predictors: In cases where the speaker's annoyance-sarcasm **congruity** was preserved, the **annoyance alignment** judgment between speakers and observers led to higher **sarcasm alignment** judgment ($\beta = 0.42, p < 0.001$). However, when this congruity was not maintained, the observer's correct identification of speaker's annoyance no longer contributed to the alignment in sarcasm judgment between speaker and observers.

To summarize, when the utterance of the speaker does not seem matched with the level of annoyance they may have felt in that context, observers are more likely to provide a sarcasm judgment that diverges from the speaker's own judgment (**H1**). If the speaker's underlying annoyance is congruous with their sarcastic utterance, the correct identification of speaker's annoyance by observers helps align observers' judgment of sarcasm with that of the speaker (**H2**). Therefore, in the next sections, based on **H1**, we conduct experiments using LLMs to examine the influence of speaker's annoyance-sarcasm **congruity** on sarcasm detection.

5 Experiment 2: Sarcasm detection by LLMs

In Section 4, we showed that speaker's annoyance-sarcasm congruity is highly correlated with sarcasm being correctly transmitted to human observers. Based on these results, we examined whether the same factor influences the sarcasm detection performance of LLMs. We conducted a classification experiment with encoder-only models and decoder-only models. The encoder-only models are classical observer models suitable for the task of sarcasm detection, and the decoder-only models are generative models that have shown their impressive capabilities to handle numerous NLP tasks. We used the fine-tuning settings for the encoder-only models, because they tend to require task-specific tuning to ensure a reasonable level of performance (Lyu et al., 2024). We used zero-shot settings for the generative models, without additional fine-tuning that requires substantial computational resources. We binary-coded the original sarcasm ratings by both sources – speakers and observers (averaged) – by using the midpoint of the scale (3.5) as the cut-off point. We downsampled CSC to have an equal number of sarcastic and non-sarcastic instances (N = 2,210 vs. 2,398).

5.1 Encoder-only models

We fine-tuned `bert-base-uncased` (110M parameters; Devlin et al., 2019) and `roberta-base` (125M parameters; Liu et al., 2019) on CSC to perform binary sarcasm classification (See Appendix A for setup details). For each language model, we obtained predictions on the test set.

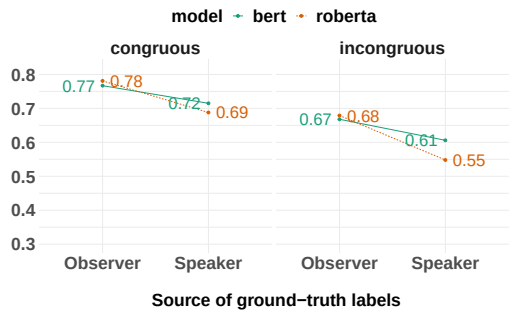


Figure 1: Macro F-scores (y-axis) for sarcasm detection by encoder-only models, according to ground-truth labels by observers or speakers. Results shown in two conditions - instances with speaker's annoyance matching the level of sarcasm (congruous) or not (incongruous).

Figure 1 shows sarcasm prediction performance by encoder-only models given the different conditions of speakers' annoyance-sarcasm congruity. In general, the instances for which speaker's annoyance level was not matching the level of sarcasm of their subsequent responses (incongruous) show lower F-scores. These results are compatible with the results about human observers described in Section 4: Cases in which sarcasm fails to be communicated to observers are related to cases in which the speaker says something that is disproportion-

ate to their emotional motivation (low annoyance-sarcasm congruity). Likewise, also for encoder-only models, sarcasm is more difficult to detect when the speaker's annoyance level is unmatched with the output utterance. We further find that these models show better detection with observer ground-truth labels than speaker ground-truth labels, which suggests their inclination to play the observer's role (Jang and Frassinelli, 2024).

5.2 Decoder-only models

We prompted smaller (≈ 3B parameters) and bigger (≈ 7-8B parameters) open-source instruction-tuned generative LLMs, in zero-shot settings[5]: `Llama3.2-3B`, `Llama3.1-8B`, `Qwen2.5-3B`, `Qwen2.5-7B`. (See Appendix B for full prompts).

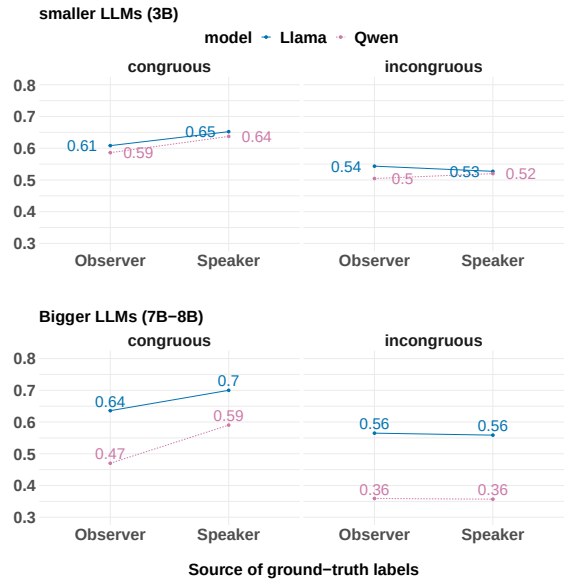


Figure 2: F-scores for sarcasm detection by generative LLMs in zero-shot settings (y-axis). Models with 3B parameters (top) or 7-8B parameters (bottom), according to ground-truth labels by observers or speakers. Results shown in two conditions - instances with speaker's annoyance matching the level of sarcasm (congruous; left) or not (incongruous; right).

Figure 2 shows sarcasm prediction performance by the generative models given different conditions of speakers' annoyance-sarcasm congruity. The best performing model is `Llama3.1-8B`, with speaker ground-truth, in the congruous condition. In general, all the generative models also struggle to detect sarcasm when the utterance is incongruous with the speaker's annoyance level, in line with the previous results from Section 5.1.

On the other hand, we see an interesting difference between the generative models and the encoder-only models. In the congruous condition, the generative models perform better with speaker ground-truth than observer ground-truth (e.g., F-scores of 0.64 vs. 0.59 for `Qwen2.5-3B` in Figure 2). This is in contrast to the results in Section 5.1, in which the encoder-only models perform better with observer ground-truth (F-scores of 0.77 vs. 0.72 for `bert-base-uncased` in Figure 1). In the incongruous condition, though, the performance of the generative models drops to about the same level between speaker ground-truth and observer ground-truth (e.g., F-score of 0.56 for observer and speaker ground-truth for `Llama3.1-8B` in Figure 2).

We observe model-specific variations as well. The `Llama3.1-8B` performs better than its smaller version `Llama3.2-3B`, whereas `Qwen2.5-7B` underperforms its smaller version `Qwen2.5-3B`. Also, between the congruous versus incongruous conditions, the performance drop by `Qwen2.5-7B` is steeper (0.59 to 0.36) than that by `Qwen2.5-3B` (0.64 to 0.52), which suggests its relatively lower robustness against speaker's incongruity.

6 Decoder-only vs. encoder-only: Identification with speaker's perspective

In Sections 6 and 7, we conduct further experiments to examine the difference between encoder-only and decoder-only models. In Section 5.2, the decoder-only generative models showed better performance with speaker ground-truth labels than observer ground-truth labels in the congruous condition. This pattern is in contrast to the pattern we observed with encoder-only models, which demonstrated better performance with labels judged by the observers (Section 5.1).

One possible explanation for such difference is that generative models are more sensitive in interpreting speakers' "point-of-view" than encoder-only models. In Section 5.2, the prompt for the generative models asking "how sarcastic is someone's response" could have biased the models to take the speaker's perspective by default. We investigated whether prompting the LLMs with more explicit instructions to take the perspective of an observer

[5]Though we also experimented with few-shot prompting, we only report results from the zero-shot experiments, as the results were comparable in both settings.

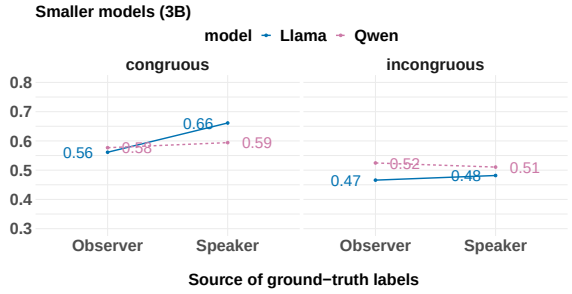


Figure 3: Macro F-scores (y-axis) for sarcasm detection by generative LLM models when explicitly prompted to take observer's perspective. Results shown in two conditions (congruous vs. incongruous).

would provide more information about why the generative LLMs perform better with speaker ground-truth than observer ground-truth (See Appendix B for the full prompt). We prompted `Llama3.2-3B` and `Qwen2.5-3B` with the new prompt.

Figure 3 shows the prediction results of `Llama3.2-3B` and `Qwen2.5-3B` explicitly prompted to take the perspective of an external observer. For both models, the prediction performance does not increase with the new prompting method. When we manually inspect the responses by the LLMs, the general tendency of these models is that they provide plenty of descriptions about the emotions that the speaker would be experiencing, or the message that the speaker is trying to express (see Appendix C for sample responses by the LLMs). Even when instructed to take the observers' perspective, the models still focus on the speaker's experience in the conversation, and use the conclusion about this as a basis to determine an external observer's sarcasm judgment.[6] Given these results, we detect a tendency that generative models identify with the speaker's perspective by default rather than observer's perspective.

7 Decoder-only vs. encoder-only: Sensitivity to disagreement among observers

In investigating the reason why generative models perform better with speaker ground-truth labels, another possibility is that generative models are more sensitive to the disagreement among multiple observers and therefore may struggle to perform at their best when the ground-truth labels are the result of simple averaging. If true, the sensitivity would have influenced the results in Section 5.2, in which the sarcasm labels by observers were averaged and binary-coded, which discards information about potential disagreement among the observers. Annotator-wide disagreement in general is considered as an important topic in NLP, as ground-truth labels cannot always come down to one single judgment (Cabitza et al., 2023; Knupleš et al., 2023; de Marneffe and Manning, 2012; Plank et al., 2014; Weber-Genzel et al., 2024). To test this, we examined whether the disagreement among the observers influences the performance of the generative LLMs more than that of the encoder-only models.

We quantified the level of disagreement among different observers using the normalized MAE described in Section 4.1. For the purpose of visual inspection, we split the scores we obtained using this formula at the mean value into *low* versus *high*. We inspected F-scores of both encoder-only models and generative models in the two groups of disagreement (low vs. high).

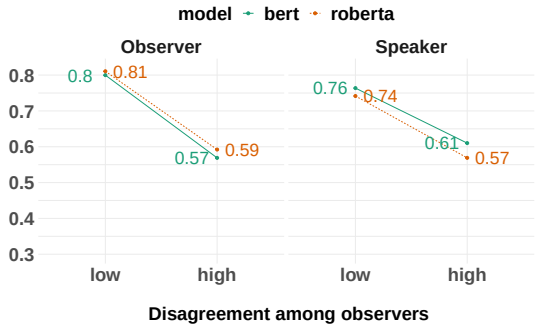


Figure 4: Macro F-scores (y-axis) for sarcasm detection by encoder-only models, per ground-truth labels by observers (left panel) and speakers (right panel). Results divided according to disagreement among observers (low vs. high).

Both encoder-only models and generative models show better performance when human annotators agree on the sarcasm judgment (Figures 4 and 5). For encoder-only models, the difference in F-score between the two groups (low vs. high) is comparable for both ground-truth labels (e.g., 0.80 vs. 0.57 in the left panel $\approx$ 0.76 vs. 0.61 in the right panel of Figure 4). In contrast, for all generative models, the difference in F-score is larger for observer ground-truth when the disagreement is low versus high (e.g., 0.71 vs. 0.53 on bottom left panel in Figure 5), than it is for speaker ground-truth

[6] Though a human observer is also expected to speculate about the speaker's emotions and communicative intentions before judging the level of sarcasm in their utterances, we think this may partially depend on the theory of mind capacity, which varies across individuals (Zhu and Wang, 2020).

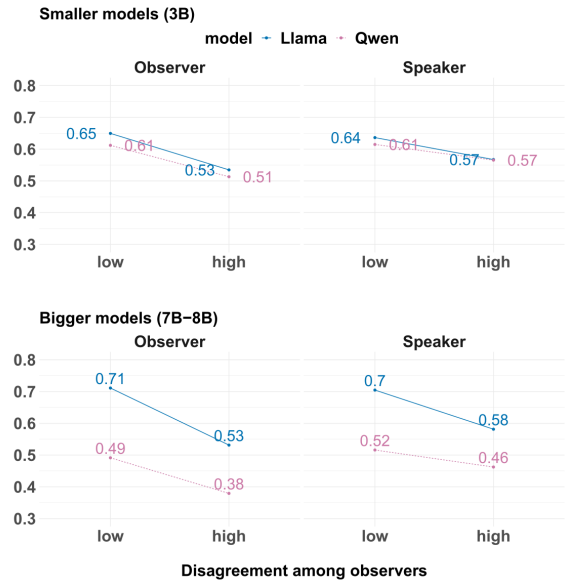


Figure 5: Macro F-scores (y-axis) for sarcasm detection by generative models, per ground-truth labels by observers (left panel) and speakers (right panel). Results divided by disagreement among observers (low vs. high).

(0.70 vs. 0.58 on bottom right panel in Figure 5). This difference is bigger for LLMs with a larger number of parameters, which suggests their higher sensitivity to disagreement among observers.

These results suggest that both types of language models are influenced by the disagreement among the observers. But in the face of this challenge, the generative models, especially those with more parameters, show a somewhat higher sensitivity by reacting against observer ground-truth to a greater extent than speaker ground-truth, which is in principle an expected behavior (i.e., in an ideal scenario, the performance with speaker ground-truth should see no change). This contrasts with encoder-only models, which show an equal drop against both sources of ground-truth labels and therefore demonstrates fragility to challenges stemming from human disagreement.

8 Experiment 3: Sarcasm detection by LLMs with additional information

Sections 4 and 5 showed that *speaker's annoyance-sarcasm congruity* influences the judgment of sarcasm both by human observers and LLMs. Here we tested whether adding information about the speaker's annoyance to LLMs would then improve the classification results.

8.1 Encoder-only models

We added information about speaker's annoyance in the form of logits to sarcasm detection models. We assessed if the added information contributes to better sarcasm detection to different degrees in congruous versus incongruous conditions. We fine-tuned `bert-base-uncased` on CSC for annoyance detection (*annoying* vs. *not annoying*). We obtained the prediction logits for annoyance on the test set, and concatenated them to the embeddings obtained from the sarcasm detection models described in Section 5.1. This concatenation strategy was inspired from the experiment in Yeo et al. (2024), which combined information about multiple dimensions into a single prediction model based on theoretical grounds. We made sure that sarcasm fine-tuning and annoyance fine-tuning would be done with the same training and test split settings to avoid the models being exposed to the same fine-tuning data for annoyance detection and sarcasm detection. We used the fine-tuned sarcasm detection models to extract embeddings as text representation, to which we added annoyance information in the form of logits. We then used a logistic regression classifier (with a 'liblinear' solver that works better for high-dimension data, and the maximum iteration of 500) on the remaining test set with a 5-fold cross-validation.

Table 3 shows the results on sarcasm classification and the improvement in performance with the addition of annoyance information. Additional annoyance information is not helpful for the encoder-only models when predicting sarcasm based on observer labels, regardless of the congruity between the sarcasm and the underlying annoyance. In contrast, when the models predict sarcasm based on speaker labels, in the incongruous condition, helping the models with additional annoyance information leads to better results (5-6%).

8.2 Decoder-only models

We prompted `Llama3.2-3B`, `Llama3.1-8B`, `Qwen2.5-3B` and `Qwen2.5-7B` with direct information about speaker's annoyance level (See Appendix B for the full prompt).

The patterns by which the added information about the speaker's annoyance helped the models varied across LLMs (See Table 3). `Qwen2.5-3B` showed comparable patterns with encoder-only models, in which adding the annoyance information increased the F-score for speaker ground-truth by a bigger margin (+0.12/+0.14) compared to ob-

		Encoder-only		Generative			
G.T	**Congruity**	**BERT**	**RoBERTa**	**Llama-3B**	**Llama-8B**	**Qwen-3B**	**Qwen-7B**
Speaker	Congruous	+0.01	+0.02	-0.04	+0.02	**+0.12**	**+0.05**
	Incongruous	**+0.05**	**+0.06**	+0.02	-0.07	**+0.14**	-0.05
Observer	Congruous	+0.00	-0.01	-0.01	+0.02	**+0.05**	+0.00
	Incongruous	+0.00	+0.00	+0.02	-0.02	**+0.07**	-0.03

Table 3: Improvement in F-score for sarcasm detection performance by different LLMs when annoyance information was additionally supplied in the form of logits (encoder-only models) and prompting (generative models). Improvement of 5% and higher marked in bold.

server ground-truth (+0.05/+0.07), and the increase being higher for incongruous condition (+0.14) than the congruous condition (+0.12). None of the other models show any consistent improvement when information about the underlying annoyance of the speakers was supplied in the prompt. We suspect that it may be because models with a larger number of parameters are less likely to be influenced by added information from one dimension only (annoyance). Nevertheless, given the inconclusive results of this experiment, further examination would be needed about the influence of assistive information for LLM performance.

9 General discussion

When speakers use sarcasm without any noticeable emotional cues, external observers lose an important source of information for judging the level of sarcasm in the provided utterance (**H1**). This tendency in human observers is also reflected in LLMs. For both humans and LLMs, sarcasm is difficult to detect when the speakers' annoyance seems unmatched with the output utterance. In contrast, in sarcastic utterances where proportional annoyance can be perceived as an underlying motivation, models are better at detecting sarcasm. Some differences are observed between encoder-only models and generative models, in terms of which source of ground-truth labels (speaker vs. observer) they match better. Encoder-only models show better performance with observer ground-truth, in line with prior work (Abu Farha et al., 2022; Jang and Frassinelli, 2024). However, generative models show better performance with speaker ground-truth. A further analysis suggests that generative models may impersonate speakers' perspective by default compared to encoder-only models. This aligns with the capabilities that these models are expected to have, exemplified by one of the evaluation suites for Llama-3 models "inhabiting a character/persona".[7]

[7]https://ai.meta.com/blog/meta-llama-3/

Nevertheless, speaker's incongruity in their underlying emotion and utterance still poses a challenge for LLMs. This is a factor worth considering for the inspection of linguistic competence of LLMs, because investigating a linguistic output by humans often requires understanding the factors that led up to it (e.g., speaker's motivation). Another obstacle that hinders good performance by LLMs is higher disagreement among observers (annotators). It is important for the evaluation of LLM capabilities to investigate the patterns by which LLMs navigate through varying linguistic judgments by humans, especially on heavily subjective topics such as sarcasm. The findings in this work also suggest that future research should address both perspectives of conversational partners (speaker vs. listener) when evaluating LLM output. Examining which perspective is reflected in the output of LLMs would help understand the competence of LLMs in more depth.

10 Conclusion

We showed that speaker's incongruity between their utterance and the annoyance they felt is associated with their judgment of sarcasm diverging from the judgment by external observers. This factor, as well as disagreement among the observers, also presented challenges to language models (both generative and encoder-only). Lastly, we discovered that the generative models are more likely to impersonate speakers' perspective more than observers' perspective, in contrast to encoder-only models.

Limitations

The main limitation of this work is that only two factors were considered as intermediary elements contributing to sarcasm failure, as mentioned in Section 3.2. We acknowledge that sarcasm can fail to be communicated for several reasons other than the mismatch between annoyance and sarcasm, given its complexity mentioned in the literature e.g., Gibbs and Colston (2023). Examining more factors

such as multimodal and other contextual factors in addressing the causes for failure of sarcastic communication is left to future work.

Another limitation of this work is that only one dataset was used for our experiments, because this is the only dataset we found to have both speaker and observer labels on multiple related dimensions (e.g., sarcasm and annoyance). Replications of our findings with other datasets and topics would strengthen our findings about communication failure in general. The use of larger language models than reported in this paper may also be considered for more conclusive insights about this topic.

Lastly, the ways of integrating annoyance information to LLMs were limited. Annoyance information was integrated with a simple concatenation of embeddings and logits for encoder-only models, and with prompting for generative models. With the results from this preliminary work, other forms of information integration will need to be tested in future work.

Acknowledgments

We are grateful to Bettina Braun for her advice on the methodology for the initial version of this work. The dissemination of this work has been supported by a grant from the Swedish Research Council (VR project 2014-39) for the establishment of the Centre for Linguistic Theory and Studies in Probability (CLASP) at the University of Gothenburg, Sweden.

Ethics Statement

We see little ethical issue related to this work. All our modeling experiments were conducted with open-source libraries, which received due citations. We did not rely on any AI-assistant tools for manuscript creation. But we acknowledge that at times, sarcasm in itself can be a sensitive topic including offensive language and content depending on the circumstances.

References

Ibrahim Abu Farha, Silviu Vlad Oprea, Steven Wilson, and Walid Magdy. 2022. SemEval-2022 task 6: iSarcasmEval, intended sarcasm detection in English and Arabic. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 802–814, Seattle, United States. Association for Computational Linguistics.

Dale J Barr, Roger Levy, Christoph Scheepers, and Harry J Tily. 2013. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of memory and language*, 68(3):255–278.

Gregory A. Bryant. 2023. Vocal strategies in verbal irony. In Raymond W. Gibbs and Herbert L. Colston, editors, *The Cambridge Handbook of Irony and Thought*, pages 197–215. Cambridge University Press, New York.

Federico Cabitza, Andrea Campagner, and Valerio Basile. 2023. Toward a perspectivist turn in ground truthing for predictive computing. 37:6860–6868.

Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. 2019. Towards multimodal sarcasm detection (an _Obviously_ perfect paper). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4619–4629, Florence, Italy. Association for Computational Linguistics.

Herbert L. Colston. 2023. Irony as social work: Opposition, expectation violation, and contrast. In Raymond W. Gibbs and Herbert L. Colston, editors, *The Cambridge Handbook of Irony and Thought*, pages 81–95. Cambridge University Press, New York.

Tanvi Dadu and Kartikey Pant. 2020. Sarcasm detection using context separators in online discourse. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 51–55, Online. Association for Computational Linguistics.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, 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.

Shelly Dews, Joan Kaplan, and Ellen Winner. 1995. Why not say it directly? The social functions of irony. *Discourse Processes*, 19(3):347–367.

Ruth Filik. 2023. Emotional responses to sarcasm. In Raymond W. Gibbs and Herbert L. Colston, editors, *The Cambridge Handbook of Irony and Thought*, pages 255–271. Cambridge University Press, New York.

Jean E. Fox Tree, J. Trevor D'Arcey, Alicia A. Hammond, and Alina S. Larson. 2020. The sarchasm: Sarcasm production and identification in spontaneous conversation. *Discourse Processes*, 57(5-6):507–533.

Raymond W. Gibbs. 2000. Irony in Talk Among Friends. *Metaphor and Symbol*, 15(1-2):5–27.

Raymond W. Gibbs and Herbert L. Colston. 2023. Irony and thought: The state of the art. In Raymond W. Gibbs and Herbert L. Colston, editors, *The Cambridge Handbook of Irony and Thought*, pages 3–14. Cambridge University Press, New York.

Hyewon Jang, Bettina Braun, and Diego Frassinelli. 2023. Intended and perceived sarcasm between close friends: What triggers sarcasm and what gets conveyed? In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 45.

Hyewon Jang and Diego Frassinelli. 2024. Generalizable sarcasm detection is just around the corner, of course! 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 4238–4249, Mexico City, Mexico. Association for Computational Linguistics.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2018. A large self-annotated corpus for sarcasm. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Urban Knupleš, Diego Frassinelli, and Sabine Schulte im Walde. 2023. Investigating the nature of disagreements on mid-scale ratings: A case study on the abstractness-concreteness continuum. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 70–86, Singapore. Association for Computational Linguistics.

Amardeep Kumar and Vivek Anand. 2020. Transformers on sarcasm detection with context. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 88–92, Online. 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*.

Yougang Lyu, Lingyong Yan, Shuaiqiang Wang, Haibo Shi, Dawei Yin, Pengjie Ren, Zhumin Chen, Maarten de Rijke, and Zhaochun Ren. 2024. KnowTuning: Knowledge-aware fine-tuning for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14535–14556, Miami, Florida, USA. Association for Computational Linguistics.

Marie-Catherine de Marneffe and Christopher D. Manning. 2012. Did it happen? the pragmatic complexity of veridicality assessment. *Computational Linguistics*, (2):301–333.

Rishabh Misra and Prahal Arora. 2023. Sarcasm detection using news headlines dataset. *AI Open*, 4:13–18.

Silviu Oprea and Walid Magdy. 2019. Exploring author context for detecting intended vs perceived sarcasm. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2859, Florence, Italy. Association for Computational Linguistics.

Silviu Oprea and Walid Magdy. 2020. iSarcasm: A dataset of intended sarcasm. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1279–1289, Online. Association for Computational Linguistics.

Shereen Oraby, Vrindavan Harrison, Lena Reed, Ernesto Hernandez, Ellen Riloff, and Marilyn Walker. 2016. Creating and characterizing a diverse corpus of sarcasm in dialogue. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 31–41, Los Angeles. Association for Computational Linguistics.

Penny M. Pexman and Kara M. Olineck. 2002. Does Sarcasm Always Sting? Investigating the Impact of Ironic Insults and Ironic Compliments. *Discourse Processes*, 33(3):199–217.

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014. Linguistically debatable or just plain wrong? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 507–511, Baltimore, Maryland. Association for Computational Linguistics.

Joan Plepi and Lucie Flek. 2021. Perceived and intended sarcasm detection with graph attention networks. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 97–105, Online. Association for Computational Linguistics.

Boaz Shmueli, Lun-Wei Ku, and Soumya Ray. 2020. Reactive Supervision: A New Method for Collecting Sarcasm Data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2553–2559, Online. Association for Computational Linguistics.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. SemEval-2018 task 3: Irony detection in English tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50, New Orleans, Louisiana. Association for Computational Linguistics.

Tony Veale. 2023. Great expectations and epic fails: A computational perspective on irony and sarcasm. In Raymond W. Gibbs and Herbert L. Colston, editors, *The Cambridge Handbook of Irony and Thought*, pages 225–234. Cambridge University Press, New York.

Leon Weber-Genzel, Siyao Peng, Marie-Catherine De Marneffe, and Barbara Plank. 2024. VariErr NLI: Separating annotation error from human label variation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2256–2269, Bangkok, Thailand. Association for Computational Linguistics.

Gerard Yeo, Shaz Furniturewala, and Kokil Jaidka. 2024. Beyond text: Leveraging multi-task learning and cognitive appraisal theory for post-purchase intention analysis. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12353–12360, Bangkok, Thailand. Association for Computational Linguistics.

Ning Zhu and Zhenlin Wang. 2020. The paradox of sarcasm: Theory of mind and sarcasm use in adults. *Personality and Individual Differences*, 163:110035.

A Fine-tuning details for Section 5.1

We used 80% of the downsampled data for fine-tuning in a 5-fold split (model initialization seeds: 10, 20). We fine-tuned each model for 2 epochs with a batch size of 16, and a learning rate of 5e-5. The fine-tuning was implemented using the Transformers class from the Hugging Face library, and was conducted on an NVIDIA A100 GPU with a total memory of 40GB. The reported results in Section 5.1 are the average of all seeds and folds.

B Prompts for generative models

The prompting was implemented using the Transformers class from the Hugging Face library, and was conducted on an NVIDIA A100 GPU with a total memory of 40GB.

B.1 Base prompt

"*Analyze each situation between Steve and John. In each situation, judge if John's response is sarcastic or not. Say 1 if John's response is sarcastic and 0 if it is not sarcastic. Only respond with 1 or 0 and nothing else: [context]. And John says [response].*".

B.2 Prompt with explicit instructions to take the observer's perspective

**We also used a prompt with the proper name of Sarah removed, and the results were comparable to using the above prompt.

"Sarah is analyzing conversations between Steve and John. In each situation, Sarah will judge if John's response is sarcastic or not. Sarah will say 1 if John's response is sarcastic and 0 if it is not sarcastic. What would Sarah say in this situation, 1 or 0?: [context]. And John says [response]."

B.3 Prompt with annoyance information

"Read each situation between Steve and John. Judge if John's response is sarcastic or not. Say 1 if John's response is sarcastic and 0 if it is not sarcastic: [context]. And John says [response]. In this situation, on a scale of 1 to 6, John's annoyance was at [annoyance-rating]".

C Reasoning provided by LLMs in Section 6

`Llama3.2-3B:`

- In this situation, Sarah would say 0. John's response seems to be a supportive and understanding comment, rather than a sarcastic one. He's acknowledging Steve's frustration and offering a sympathetic ear, which suggests that John is not being sarcastic.

- Based on the conversation, Sarah would say 1. Although John's response seems to be a straightforward explanation ("it's just because I want to pick your girlfriend up later!"), the initial response ("come on, why can't I borrow it? Don't you trust me?") has a tone of sarcasm and implies that John is questioning Steve's trust, which is a common characteristic of sarcastic responses.

- In this situation, Sarah would say 0. Although John's response seems to be slightly negative towards social media, it does not appear to be sarcastic. John's statement "Life's too short" is a genuine expression of his opinion, and his offer to help Steve with posting a picture is not sarcastic.

`Qwen2.5-3B:`

- In this scenario, John's statement "In about an hour" does not appear to be sarcastic. Steve's initial response indicates that he is genuinely interested in joining the party and wants to know when it starts. John's reply simply confirms the timing of the party without any indication of sarcasm. Therefore, Sarah would say: 0.

- In this scenario, John's response seems genuine rather than sarcastic. John is complimenting the watch and even going so far as to suggest that it has become even more shiny since he last saw it. This implies that he finds the watch impressive and possibly in better condition now. Therefore, Sarah would judge this response as **0** (not sarcastic).

- In this situation, John's response can be interpreted as somewhat sarcastic. John acknowledges Steve's offer to lend him a shirt but then suggests he already has another shirt that isn't

tight. This could imply that John feels his current shirt is tight and that he doesn't need to accept the offer, which might be seen as a subtle rejection or at least an attempt to downplay Steve's offer. Given this context, Sarah would likely judge John's response as sarcastic. So, Sarah would say **1**.

A Model of Information State in Situated Multimodal Dialogue

Kenneth Lai[1], **Lucia Donatelli**[2], **Richard Brutti**[1], **James Pustejovsky**[1]
[1]Brandeis University, Waltham, MA, USA
[2]Vrije Universiteit Amsterdam, Amsterdam, Netherlands
{klai12, brutti, jamesp}@brandeis.edu, l.e.donatelli@vu.nl

Abstract

In a successful dialogue, participants come to a mutual understanding of the content being communicated through a process called *conversational grounding*. This can occur through language, and also via other communicative modalities like gesture. Other kinds of actions also give information as to what has been understood from the dialogue. Moreover, achieving common ground not only involves establishing agreement on a set of facts about discourse referents, but also agreeing on what those entities refer to in the outside world, i.e., *situated grounding*. We use examples from a corpus of multimodal interaction in a task-based setting, annotated with Abstract Meaning Representation (AMR), to explore how speech, gesture, and action contribute to the construction of common ground. Using a simple model of information state, we discuss ways in which existing annotation schemes facilitate this analysis, as well as information that current annotations do not yet capture. Our research sheds light on the interplay between language, gesture, and action in multimodal communication.

1 Introduction

In dialogue, the concept of *common ground* refers to the set of presuppositions held by the participants, propositions that they agree to treat as true (Stalnaker, 1978). The process by which common ground is constructed over a dialogue is known as *(conversational) grounding* (Clark and Brennan, 1991). Formal models of dialogue have been developed to track how common ground (and more generally, information state) evolves over the course of an interaction (Poesio and Traum, 1997; Cooper and Larsson, 1999; Ginzburg, 2012).

Much work examining the role of non-linguistic modalities in communication focuses on gesture (Kendon, 2004; McNeill, 2008; Lascarides and Stone, 2009). This includes analyses of the semantic contents of gestures (Ebert and Ebert, 2014; Schlenker, 2018), and proposals for integrating gesture into models of dialogue (Lücking and Ginzburg, 2020).

Figure 1: Example of multimodal communication in a task-based setting (Wang et al., 2017). On the left, the signaler describes part of the structure to be built: he says, "It starts in the top left; there's a block", and makes a deictic gesture with his left hand. On the right, the actor puts a block in the top left corner of the table (note that both videos are mirrored).

More general types of actions can also affect dialogue context, especially in real-world or embodied settings (Tam et al., 2023). Within these settings, *referential grounding* is the process by which interlocutors anchor linguistic expressions to actual entities, relations, or events in the shared environment. When considering the perception and embodiment of participants, *situated grounding* is used (Kordjamshidi et al., 2025). In other words, while conversational grounding focuses on "what was said", referential grounding ensures everyone agrees on "what is being talked about".

In this paper, we present a simple information state model of dialogue that integrates both propositional updates (conversational grounding) and referential anchoring (situated grounding). We walk through a dialogue fragment from a corpus of task-based multimodal interaction (Lai et al. (2024); Wang et al. (2017); an example is shown in Fig-

Proceedings of the 16th International Conference on Computational Semantics, pages 292–298
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

ure 1), annotated with AMR (Banarescu et al., 2013) for speech and gesture (Brutti et al., 2022; Donatelli et al., 2022), illustrating how speech, gesture, and object-directed actions co-construct and update the common ground. We assess the strengths and limitations of current annotations for capturing multimodal grounding phenomena, and argue for the importance of situational information in dialogue interpretation.

2 Related Work

Information state theories of dialogue are based on the idea that dialogue acts change the context available to participants (Fernández, 2022). At the most basic level, this includes the common ground, or shared assumptions of the participants (Stalnaker, 1978). Over time, the scope of the information state has expanded to handle different types of utterances beyond assertion; interrogatives are commonly handled via a set or stack of questions under discussion (Roberts, 2012), while there are various theories for the meaning of imperatives (Kaufmann, 2012; Portner, 2004; Barker, 2012). Formal information state theories include Poesio and Traum (1997); Cooper and Larsson (1999); and Ginzburg (2012).

Several dialogue corpora analyze the conversational grounding process and the impact of situated grounding or information about the shared environment. Among them, Mohapatra et al. (2024) annotate two corpora with (conversational) grounding acts and grounding units (Traum, 1995). The STAC corpus contains multi-party Settlers of Catan chats annotated with discourse structure and dialogue acts (Asher et al., 2016); Martinenghi et al. (2024) experiment with using large language models to predict the dialogue acts. Zhu et al. (2023) present the FIREBALL dataset of Dungeons & Dragons games, showing that adding game state information to the dialogue history can improve narration generation. Kruijt et al. (2024) develop the SPOTTER framework to investigate linguistic convention formation in a task referentially grounded in vision. The SCOUT corpus of situated human-robot dialogues (Lukin et al., 2024) is annotated with Dialogue-AMR (Bonial et al., 2020) and relations between utterances (Carletta et al., 1996; Traum et al., 2018). The Weights Task Dataset of situated interaction is annotated with several modalities including speech and gesture (Khebour et al., 2024a); Khebour et al. (2024b) perform common ground tracking, focusing on the emergence of facts.

3 Analyzing Multimodal Interaction

3.1 Setting

We draw examples in this paper from the EGGNOG corpus of task-based multimodal communication (Wang et al., 2017). Two participants are located in separate rooms, connected through video and audio. One person, the signaler, has an image of a block structure, and instructs the other person, the actor, on how to build the structure. For part of the corpus, Lai et al. (2024) annotated the signaler's speech and gesture with AMR (Banarescu et al., 2013; Brutti et al., 2022; Donatelli et al., 2022). While they did not annotate the actor's actions, our examples use another AMR extension, Action AMR (Tam et al., 2023), to describe them.

3.2 Information State

We use a simple model of information state, inspired by Ginzburg (2012)'s dialogue gameboard. Our model $M = (C, Q, T_s, T_a, E, g)$ contains the common ground C, which we assume to have a similar structure to a file card (Heim, 1982) or Discourse Representation Structure (Kamp, 2002), namely, that it stores a set of discourse referents and facts or shared beliefs about them. It also contains a set of questions under discussion Q. We take imperatives to denote actions; while Barker (2012) does not prescribe any specific data structure for these, we adopt Portner (2004)'s concept of a To-Do List T (one each for the signaler s and actor a) to handle actions. To describe the environment in which the participants are situated, we use a list E containing the objects in the environment (including the agents themselves), and the previous actions performed, both communicative and not; this is similar to the "common ground structure" in Pustejovsky and Krishnaswamy (2021) and Lai et al. (2021). Finally, to represent the situated grounding of objects and actions to the environment, we use an embedding or grounding function g. This is similar to the notion of an embedding in Discourse Representation Theory (Kamp, 2002), a function mapping discourse referents to elements in a model; here, the "model" comprises the environment E in which the agents are situated. For simplicity, we assume that the information state is an objective structure (i.e., not relative to any particular agent), and that all of its components are public; while each agent is assigned their own To-Do List, they also have access to the other participant's list.

Figure 2: Initial state for our example.

3.3 Example

We illustrate the dynamics of our information state using an example from the corpus. We note that because of the task-based nature of the interaction, the state does not begin empty. Both participants have prior information about the task, given by the experimenter or from previous trials; the common ground begins with these task-based presuppositions (see additional discussion in Section 4). Similarly, "what is the shape of the structure?" can be seen as an overarching question that begins in Q, the signaler has the task of communicating how to build the structure in T_s, and the actor has the task of actually building the structure in T_a. The environment contains the participants, the actor's table[1], and the blocks, at least[2]. Finally, our example begins with the signaler already having given one instruction and the actor having put a block on the table, as shown in Figure 2.

In the corpus, signalers generally communicate their instructions through a combination of direct commands, and/or describing some aspect of the eventual structure. Here, the signaler does the former, issuing the imperative "Take another block; put it next to it" and gesturing towards a location on his table, as shown in Figure 3 (an example of the latter follows in Section 4). The signaler's communicative act adds discourse referents to the common ground and actions to the actor's To-Do List; the communicative act is itself recorded in

[1]The signaler and actor being in different rooms complicates things somewhat. The signaler and actor both have tables in their rooms, and the signaler often uses locations on their table to refer to locations on the actor's table, raising interesting questions of perspective and frame of reference. Ultimately, the actor's table and the locations on it are the ones relevant to the completion of the task.

[2]One could argue that the environment should also include the locations in space available to the participants. Assuming a continuous space, enumerating every possible location would not be possible, so we allow for actions to dynamically generate locations as needed, a strategy employed by Krishnaswamy and Pustejovsky (2021).

(1) "Take another block; put it next to it."

```
(a / and
  :op1 (t / take-01 :mode imperative
          :ARG0 (y / you)
          :ARG1 (b / block
                   :mod (a2 / another)))
  :op2 (p / put-01 :mode imperative
          :ARG0 y
          :ARG1 b
          :ARG2 (n / next-to
                   :op1 (i / it))))
```

(2) Gesture for "put here".

```
(g / gesture-unit
  :op1 (d / deixis-GA
          :ARG0 (s / signaler)
          :ARG1 (l / location)
          :ARG2 (a / actor))
  :op2 (i / icon-GA
          :ARG0 s
          :ARG1 (p / put-01)
          :ARG2 a))
```

Figure 3: The signaler gives the actor an instruction using speech (1) and gesture (2). Colors denote coreference relations between the AMRs.

the environment. These discourse referents and actions come from the speech and gesture AMRs, also shown in Figure 3. In this case, the signaler references a new `block` `b` to be placed at a new `location` `l`, and places `take` (`t`) and `put` (`p`) actions into T_a.

The actor shows her understanding of the signaler's instructions by performing the referenced actions. The action and its corresponding AMR are shown in Figure 4. In the action AMR, note that the action and its arguments are not discourse objects, but rather objects in the world, that is, they are elements of E; for clarity, we use capital letters in the action AMR to mark this distinction. In performing the action, the actor *identifies* entities in the discourse with entities in the world, and *proposes* this identification to the signaler. That is, she is suggesting that $g(\texttt{b}) = \texttt{B2}$, $g(\texttt{l}) = \texttt{L2}$, $g(\texttt{p}) = \texttt{P2}$, and (given a suitably subevent structure for `put`, such as in Krishnaswamy and Pustejovsky (2021)), $g(\texttt{t})$ is a subevent of `P2`.

Note that the actor's action does not automati-

(3) Actor puts another block next to the first block.

```
(P2 / put-01
  :ARG0 (A / actor)
  :ARG1 (B2 / block)
  :ARG2 (L2 / location))
```

Figure 4: The actor carries out the signaler's instruction. Proposed situated grounding between the action AMR and the communicative act is shown with the same colors as above (a subevent of the actor's `put` action corresponding to the signaler's `take` instruction).

cally update the situated grounding function g; it is now up to the signaler to accept or reject the actor's proposals. Mirroring Ginzburg (2012)'s treatment of statements yet to be accepted, the actor's suggestions become questions under discussion, $(g(\texttt{b}) = \texttt{B2})?$, and so on. If the signaler is satisfied with the actor's action, they can either give explicit positive acknowledgment, or implicitly accept by moving on to the next instruction; either way it is the signaler's acceptance that updates g. Otherwise, if there is something wrong, the signaler can either say or gesture so, and/or provide additional instruction to correct the misunderstanding.

In this example, the speaker's next communicative act is the utterance "Spread them apart a little bit but not as wide as a full block", with a corresponding "spread apart" gesture. While the actor's choice of block may have been appropriate, and $(g(\texttt{b})$ is thus set to $\texttt{B2})$, the signaler intended there to be a gap between the blocks, and the actor's proposal of $(g(\texttt{l}) = \texttt{L2})?$ is *not* accepted. The actor responds by moving both blocks to new locations a suitable distance apart; this represents a proposal not only to set the location of the second block, but also to update the location of the first block. The new proposals are eventually accepted by the signaler, and the dialogue continues.

4 Discussion and Conclusion

Within the corpus, some signalers use what we can call the *result present tense*, describing the configuration resulting from an action in the present, rather than giving an imperative. In fact, excluding one-word utterances, declarative sentences outnumber imperatives by almost two to one (191 to 97). In one example, the signaler says "Starting from the top, moving to your left, down four diagonally a row with the corners touching." The analysis of such utterances can be formalized in a number of ways. One approach, suggestive of Ross (1970)'s performative analysis, is to treat them like implicit imperatives: one could imagine each statement beginning with a covert "Make it true that...". These instructions would then be added to the actor's To-Do List, in the same way as explicit imperatives[3]. Another approach is to treat them as standard declaratives, with the actor's subsequent actions determined by pragmatic effects. Following Ginzburg (2012), declarative statements are offered as questions under discussion, which the actor can either accept or reject. Without an imperative, there is no direct update to the actor's To-Do List; however, assuming that they accept the statement, and the initial overarching task of building the structure remains in T_a, they will change the state of the world (i.e., move blocks around) to make the signaler's description true.

The challenge of ambiguous statements that require context for correct interpretation are well-established in dialogue literature (Grice, 1975). In sampling our corpus, we encounter two distinct kinds of ambiguity that require situated information to arrive at the correct interpretation. First, we notice several instances of presuppositions that are connected to the setup of the block-building task. These presuppositions are triggered with canonical utterances such as "again", "the same", or "also" (Frege, 1892; Strawson, 1950; Stalnaker, 1975). In one interaction, the signaler begins with the statement, "so you will begin with a grid structure **again**", referencing a previous interaction that required a grid-like spatial understanding of the block orientation on the table. We notice this throughout interactions: both signalers and actors approach the task with an implicit and often shared understanding of constraints on block structures and their orientation in the physical space.

In the same interaction, the signaler instructs the actor to create "**the same** pattern" with blocks in a new area of the table. Here, we encounter a second, partially overlapping challenge of multimodal ambiguity: multimodal coreference. In the case of the block pattern, the instruction and subsequent action

[3]We thank an anonymous reviewer for this suggestion.

are potential instances of the so-called *sloppy identity* effect (Ross, 1967), in which the same phrase can be interpreted with different arguments, i.e., blocks (Partee, 1975; Webber, 1978; Carnie, 2021). Such multimodal coreference can also be understood as *coreference under transformation* (Rim et al., 2023), a category easier to annotate and helpful in understanding sequences of events. Here, while the concept of a block pattern is stable in identity, the concept is applied to a new instance that requires situated knowledge to enact correctly.

Using AMR for both speech and gesture allows multimodal coreference relations throughout the dialogue and between the modalities to be marked using Multi-sentence AMR (O'Gorman et al., 2018). Meanwhile, using AMR for action facilitates alignment and binding from the communicative modalities to the local environment, allowing for easier identification of situated grounding. However, as the Lai et al. (2024) corpus annotates only communication from the signaler, there are certain aspects of conversational grounding, such as the signaler's understanding of the actor's communicative acts, that the annotations do not capture yet. A complete analysis of bidirectional grounding processes will require the rest of the corpus to be annotated with the actor's actions, in addition to their speech and gesture. Our model, focusing on describing *what* identifications are made between discourse entities and objects in the real world, sidesteps the question of *how* agents make these identifications. Kennington and Schlangen (2015) describe a "words as classifiers" approach to situated grounding of words and phrases in perceptual scenes. Furthermore, our findings are limited to a single corpus, and applying this approach to other dialogue types will reveal new insights. For example, in the block structure-building task, the signaler knows what structure is to be built, and the actor knows this, and therefore accepts the signaler as an authoritative source of information. Additionally, the task-specific presuppositions that define the initial dialogue state require knowledge of each new context. These factors point to clear next steps for extending multimodal semantic annotation for the analysis of situated dialogue.

Acknowledgments

We would like to thank Derrick Kim and Yifan Zhu for their assistance with this research. We would also like to thank the three anonymous reviewers for their detailed comments and suggestions.

This research was supported by the NSF National AI Institute for Student-AI Teaming (iSAT) under grant DRL 2019805 to James Pustejovsky. The opinions expressed are those of the authors and do not represent views of the NSF.

References

Nicholas Asher, Julie Hunter, Mathieu Morey, Benamara Farah, and Stergos Afantenos. 2016. Discourse structure and dialogue acts in multiparty dialogue: the STAC corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2721–2727, Portorož, Slovenia. European Language Resources Association (ELRA).

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.

Chris Barker. 2012. Imperatives denote actions. In *Proceedings of Sinn und Bedeutung*, volume 16, pages 57–70.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie M. Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. 2020. Dialogue-AMR: Abstract Meaning Representation for dialogue. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 684–695, Marseille, France. European Language Resources Association.

Richard Brutti, Lucia Donatelli, Kenneth Lai, and James Pustejovsky. 2022. Abstract Meaning Representation for gesture. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1576–1583, Marseille, France. European Language Resources Association.

Jean Carletta, Amy Isard, Stephen Isard, Jacqueline Kowtko, Gwyneth Doherty-Sneddon, and Anne Anderson. 1996. *HCRC dialogue structure coding manual*. Human Communication Research Centre.

Andrew Carnie. 2021. *Syntax: A generative introduction*. John Wiley & Sons.

Herbert H. Clark and Susan E. Brennan. 1991. Grounding in communication. In Lauren B. Resnick, John M. Levine, and Stephanie D. Teasley, editors, *Perspectives on Socially Shared Cognition*, pages 127–149. American Psychological Association.

Robin Cooper and Staffan Larsson. 1999. Dialogue moves and information states. In *Proceedings of*

the Third International Workshop on Computational Semantics.

Lucia Donatelli, Kenneth Lai, Richard Brutti, and James Pustejovsky. 2022. Towards situated AMR: Creating a corpus of gesture AMR. In *Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management. Health, Operations Management, and Design*, pages 293–312, Cham. Springer International Publishing.

Cornelia Ebert and Christian Ebert. 2014. Gestures, demonstratives, and the attributive/referential distinction. *Handout of a talk given at Semantics and Philosophy in Europe (SPE 7), Berlin*, 28.

Raquel Fernández. 2022. Dialogue. In *The Oxford Handbook of Computational Linguistics*. Oxford University Press.

Gottlob Frege. 1892. On sense and reference.

Jonathan Ginzburg. 2012. *The Interactive Stance*. Oxford University Press.

H. Paul Grice. 1975. Logic and conversation. In Donald Davidson, editor, *The logic of grammar*, pages 64–75. Dickenson Pub. Co.

Irene Heim. 1982. *The Semantics of Definite and Indefinite Noun Phrases*. Ph.D. thesis, University of Massachusetts Amherst.

Hans Kamp. 2002. A theory of truth and semantic representation. In Paul H. Portner and Barbara H. Partee, editors, *Formal Semantics - the Essential Readings*, pages 189–222. Blackwell.

Magdalena Kaufmann. 2012. *Interpreting Imperatives*. Springer Netherlands, Dordrecht.

Adam Kendon. 2004. *Gesture: Visible Action as Utterance*. Cambridge University Press.

Casey Kennington and David Schlangen. 2015. Simple learning and compositional application of perceptually grounded word meanings for incremental reference resolution. 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 292–301, Beijing, China. Association for Computational Linguistics.

Ibrahim Khebour, Richard Brutti, Indrani Dey, Rachel Dickler, Kelsey Sikes, Kenneth Lai, Mariah Bradford, Brittany Cates, Paige Hansen, Changsoo Jung, Brett Wisniewski, Corbyn Terpstra, Leanne Hirshfield, Sadhana Puntambekar, Nathaniel Blanchard, James Pustejovsky, and Nikhil Krishnaswamy. 2024a. When text and speech are not enough: A multimodal dataset of collaboration in a situated task. *Journal of Open Humanities Data*.

Ibrahim Khalil Khebour, Kenneth Lai, Mariah Bradford, Yifan Zhu, Richard A. Brutti, Christopher Tam, Jingxuan Tu, Benjamin A. Ibarra, Nathaniel Blanchard, Nikhil Krishnaswamy, and James Pustejovsky. 2024b. Common ground tracking in multimodal dialogue. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3587–3602, Torino, Italia. ELRA and ICCL.

Parisa Kordjamshidi, Marie-Francine Moens, and James Pustejovsky. 2025. *Spatial Language Understanding: Representation, Reasoning, and Grounding*. Springer - Synthesis Lectures on Human Language Technologies, Switzerland.

Nikhil Krishnaswamy and James Pustejovsky. 2021. The role of embodiment and simulation in evaluating hci: Experiments and evaluation. In *International Conference on Human-Computer Interaction*, pages 220–232. Springer.

Jaap Kruijt, Peggy van Minkelen, Lucia Donatelli, Piek T.J.M. Vossen, Elly Konijn, and Thomas Baier. 2024. SPOTTER: A framework for investigating convention formation in a visually grounded human-robot reference task. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15202–15215, Torino, Italia. ELRA and ICCL.

Kenneth Lai, Richard Brutti, Lucia Donatelli, and James Pustejovsky. 2021. Situated umr for multimodal interactions. In *Proceedings of the 25th Workshop on the Semantics and Pragmatics of Dialogue - Poster Abstracts*, Potsdam, Germany. SEMDIAL.

Kenneth Lai, Richard Brutti, Lucia Donatelli, and James Pustejovsky. 2024. Encoding gesture in multimodal dialogue: Creating a corpus of multimodal AMR. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5806–5818, Torino, Italia. ELRA and ICCL.

Alex Lascarides and Matthew Stone. 2009. A Formal Semantic Analysis of Gesture. *Journal of Semantics*, 26(4):393–449.

Andy Lücking and Jonathan Ginzburg. 2020. Towards the score of communication. In *Proceedings of the 24th Workshop on the Semantics and Pragmatics of Dialogue - Full Papers*, Virtually at Brandeis, Waltham, Massachusetts, USA. SEMDIAL.

Stephanie M. Lukin, Claire Bonial, Matthew Marge, Taylor A. Hudson, Cory J. Hayes, Kimberly Pollard, Anthony Baker, Ashley N. Foots, Ron Artstein, Felix Gervits, Mitchell Abrams, Cassidy Henry, Lucia Donatelli, Anton Leuski, Susan G. Hill, David Traum, and Clare Voss. 2024. SCOUT: A situated and multimodal human-robot dialogue corpus. In *Proceedings*

of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 14445–14458, Torino, Italia. ELRA and ICCL.

Andrea Martinenghi, Gregor Donabauer, Simona Amenta, Sathya Bursic, Mathyas Giudici, Udo Kruschwitz, Franca Garzotto, and Dimitri Ognibene. 2024. LLMs of catan: Exploring pragmatic capabilities of generative chatbots through prediction and classification of dialogue acts in boardgames' multiparty dialogues. In *Proceedings of the 10th Workshop on Games and Natural Language Processing @ LREC-COLING 2024*, pages 107–118, Torino, Italia. ELRA and ICCL.

David McNeill. 2008. *Gesture and Thought*. University of Chicago Press.

Biswesh Mohapatra, Seemab Hassan, Laurent Romary, and Justine Cassell. 2024. Conversational grounding: Annotation and analysis of grounding acts and grounding units. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3967–3977, Torino, Italia. ELRA and ICCL.

Tim O'Gorman, Michael Regan, Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. 2018. AMR beyond the sentence: the multi-sentence AMR corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3693–3702, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Barbara Partee. 1975. Montague grammar and transformational grammar. *Linguistic Inquiry*, 6(2):203–300.

Massimo Poesio and David R. Traum. 1997. Conversational actions and discourse situations. *Computational Intelligence*, 13(3):309–347.

Paul Portner. 2004. The semantics of imperatives within a theory of clause types. In *Semantics and Linguistic Theory*, pages 235–252.

James Pustejovsky and Nikhil Krishnaswamy. 2021. Embodied human computer interaction. *KI - Künstliche Intelligenz*, 35(3):307–327.

Kyeongmin Rim, Jingxuan Tu, Bingyang Ye, Marc Verhagen, Eben Holderness, and James Pustejovsky. 2023. The coreference under transformation labeling dataset: Entity tracking in procedural texts using event models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12448–12460, Toronto, Canada. Association for Computational Linguistics.

Craige Roberts. 2012. Information structure in discourse: Towards an integrated formal theory of pragmatics. *Semantics and Pragmatics*, 5(6):1–69.

John Robert Ross. 1967. *Constraints on Variables in Syntax*. Ph.D. thesis, MIT.

John Robert Ross. 1970. On declarative sentences. In Roderick A. Jacobs and Peter S. Rosenbaum, editors, *Readings in English Transformational Grammar*, pages 222–272. Georgetown University Press, Washington, DC, USA.

Philippe Schlenker. 2018. Gesture projection and cosuppositions. *Linguistics and Philosophy*, 41(3):295–365.

Robert Stalnaker. 1975. Presuppositions. In *Contemporary Research in Philosophical Logic and Linguistic Semantics: Proceedings of a Conference Held at the University of Western Ontario, London, Canada*, pages 31–41. Springer.

Robert Stalnaker. 1978. Assertion. *Syntax and Semantics*, 9:315–332.

Peter F Strawson. 1950. On referring. *Mind*, 59(235):320–344.

Christopher Tam, Richard Brutti, Kenneth Lai, and James Pustejovsky. 2023. Annotating situated actions in dialogue. In *Proceedings of the Fourth International Workshop on Designing Meaning Representations*, pages 45–51, Nancy, France. Association for Computational Linguistics.

David Traum, Cassidy Henry, Stephanie Lukin, Ron Artstein, Felix Gervits, Kimberly Pollard, Claire Bonial, Su Lei, Clare Voss, Matthew Marge, Cory Hayes, and Susan Hill. 2018. Dialogue structure annotation for multi-floor interaction. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

David Rood Traum. 1995. *A computational theory of grounding in natural language conversation*. Ph.D. thesis, University of Rochester, USA. UMI Order No. GAX95-23171.

Isaac Wang, Mohtadi Ben Fraj, Pradyumna Narayana, Dhruva Patil, Gururaj Mulay, Rahul Bangar, J. Ross Beveridge, Bruce A. Draper, and Jaime Ruiz. 2017. EGGNOG: A continuous, multi-modal data set of naturally occurring gestures with ground truth labels. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 414–421.

Bonnie Lynn Webber. 1978. *A Formal Approach to Discourse Anaphora*. Routledge.

Andrew Zhu, Karmanya Aggarwal, Alexander Feng, Lara J. Martin, and Chris Callison-Burch. 2023. FIREBALL: A dataset of dungeons and dragons actual-play with structured game state information. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4171–4193, Toronto, Canada. Association for Computational Linguistics.

Learning to Refer: How Scene Complexity Affects Emergent Communication in Neural Agents

Dominik Künkele[1] and **Simon Dobnik**[1,2]
Department of Philosophy, Linguistics and Theory of Science[1]
Centre for Linguistic Theory and Studies in Probability (CLASP)[2]
University of Gothenburg, Sweden
contact@dominik-kuenkele.de and simon.dobnik@gu.se

Abstract

We explore how neural network-based agents learn to map continuous sensory input to discrete linguistic symbols through interactive language games. One agent describes objects in 3D scenes using invented vocabulary; the other interprets references based on attributes like shape, color, and size. Learning is guided by feedback from successful interactions. We extend the CLEVR dataset with more complex scenes to study how increased referential complexity impacts language acquisition and symbol grounding in artificial agents.

1 Introduction and Background

How do cognitive systems bridge the gap between rich, continuous sensory experiences and the sparse, discrete symbols used in communication? While perception operates through continuous signals, linguistic communication relies on finite vocabularies that must ground meaning about the perceived world (Regier, 1996; Roy, 2005; Cooper, 2023). This representational challenge, known as the symbol grounding problem (Harnad, 1990), becomes particularly acute in artificial systems where discrete symbols must acquire meaning through interaction rather than pre-programmed associations.

Referring expressions require systems to map visual attributes onto linguistic descriptions that uniquely identify target objects and thus can be used to study symbol grounding. Dale and Reiter (1995) formalized this process through an incremental generation algorithm that constructs descriptions by systematically adding distinguishing properties in order of salience until achieving unique identification. By this, referring expression only contain attributes that are necessary to discriminate the target from the surroundings.

Research in this area investigates how artificial agents can develop referential abilities through language games - interactive scenarios where communication protocols emerge from repeated coordination attempts (Clark, 1996; Bartlett and Kazakov, 2005; Kirby et al., 2008; Steels and Loetzsch, 2009; Kharitonov et al., 2019; Lazaridou et al., 2017). Modern implementations use deep neural networks as agents that exchange discrete messages to solve visual discrimination tasks, allowing systematic study of how symbol meaning emerges from interaction.

This paper examines emergent referential communication in neural agents tasked with identifying objects in 3D visual scenes. Using a highly controlled extension of the CLEVR dataset (Johnson et al., 2017a), we are able to manipulate the bias the neural agents are able to use in the emergent communication. We are able to vary the complexity of referential scenarios to understand the constraints governing successful symbol grounding. Our work is a study of how increasing the complexity of the scene (and therefore the space of potential referential expressions to be learned) affects learning through interaction of particular configurations of neural networks.

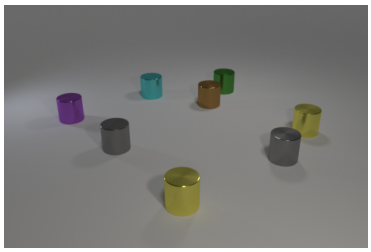
2 Dataset

We extend the original CLEVR framework (Johnson et al., 2017a) to have more control over the generated scenes.[1] By this, the objects in the generated images are controlled to have different human-recognizable attributes, namely the *shape*, *size* and *color*. These attributes also correspond to referring expressions in natural language such as English which effectively biases the agents to learn a language that is comparable to a human language.

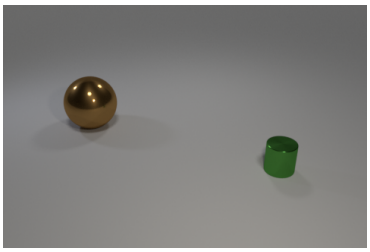
The objects in the scene are separated into two categories: one *target object* and a controlled num-

[1] github.com/DominikKuenkele/MLT_Master-Thesis_clevr-dataset-gen

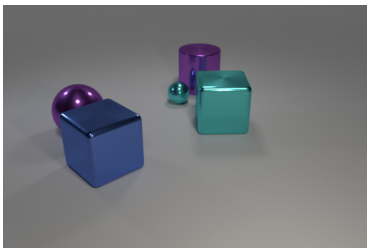
Proceedings of the 16th International Conference on Computational Semantics, pages 299–307
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License



(a) 'CLEVR color', *small brown cylinder*



(b) 'Dale-2', *small green cylinder*



(c) 'Dale-5', *large purple cylinder*

Figure 1: Example images of each dataset, with the target object specified.

ber of *distractor* objects. The target object is the main object in the scene and the models are trained to identify and communicate it between each other. This object is unique in the scene in respect to the attributes. The distractor group contains objects that can share a maximum of two attributes per object. Distractors are not required to be unique.

Using these rules, we generate three datasets with the following constraints: (i) the size of the generated images is 480×320 pixels; (ii) 10.000 images are created for each of the datasets; (iii) each image contains a maximum of 10 objects, that are not intersecting, have the same minimum distance between objects and are at least partially visible from the camera.

2.1 CLEVR color

The first generated dataset is called 'CLEVR color', in which the target object is identifiable by just the color. Both *shape* and *size* of all distractors are shared with the target object. The distractor group can contain in between 6 and 9 objects.

As seen in Figure 1(a), the *small brown cylinder* is unique. By this, it is possible to refer to the target object using the attributes with four different combinations: the *brown* object, the *brown cylinder*, the *small brown* object and the *small brown cylinder*.

2.2 CLEVR Dale datasets

The above described dataset is very restrictive in the relation between the objects, where only *one* attribute is used to disambiguate them. The number and the type of shared attributes are controlled exactly. In the real world, objects have overlapping attributes and hence objects can often be identified by an intersection of multiple attributes. For this, we created a dataset that allows almost any relation between a target object and the distractors. The creation is inspired by the incremental algorithm for the Generation of Referring Expressions (GRE) described in (Dale and Reiter, 1995) who observe that attributes in descriptions occur in certain order and are added incrementally in a certain hierarchy. This algorithm ensures that every scene contains a unique object in respect to its and the distractors' attributes. Using the algorithm, one can refer to an object using its attributes to discriminate it from all other objects as efficiently as possible. In other words, the object is described unambiguously using the lowest number of words. On the other side, it is not controlled which attributes are shared; they are assigned randomly.

Two datasets following these rules are created. The 'Dale-2' dataset contains one target object and one distractor (see Figure 1(b)), while the Dale-5 dataset contains one target object and exactly four distractors. Consider Figure 1(c), with the target object being the *large purple cylinder*. The large purple sphere shares the size and color, the two cubes only share the size, and the small turquoise sphere doesn't share any attribute.

3 Method

3.1 Image processing

To extract the features and process the images of the datasets, we build upon the proposed architecture in Johnson et al. (2017b) which was used to train baseline models on the original CLEVR dataset. Hereby, the image is first passed through a frozen ResNet-101 model (He et al., 2016). Two convolutional layers with subsequent *ReLU* nonlinearities condense the important information from the output of the feature extractor. The convolutional layers reduce the channels to 128 channels, using a kernel size of 3 and a stride and padding of 1. This matrix represents the encoded image with its extracted features.

3.2 Language Games

The goal of this research is to run and compare different setups of language games systematically. To do this, all experiments rely on the ***E**mergence of lan**G**uage in **G**ames* (EGG) framework (Kharitonov et al., 2019). This framework allows the implementation of language games in code, where two neural models agents communicate through a unidirectional discrete channel. A sender agent processes visual input. The result is used as the initial hidden state for the encoder LSTM. This LSTM is then producing symbols until it generates an <eos> symbol. The receivers' decoder LSTM processes the message symbol by symbol with a randomly initialized hidden state. After each time, a symbol is processed by the LSTM, the resulting new hidden state is passed to the receiver's neural model as the parsed message. The receiver agent is combining it with its representation of the image input and is predicting an output. In other words the receiver agent produces as many outputs as symbols are present in the message. The loss is calculated for each of these outputs separately. These losses are summed up to a total loss that is used to adapt the weights in both agents as well as in both LSTMs. As the discrete sampled categorical distribtion of the message can't be differentiated, we use Gumbel-Softmax relaxation (Jang et al., 2017) to turn it into a continuous distribution, thus allowing backpropagation through the whole language game.

4 Experiments

4.1 Attending in a language game

Setup

Two agents are tasked to solve a referring problem together. The receiver needs to 'point' to the target object in the visual scene that the sender is describing. However, only the sender is aware of which of the shown objects is the target object. To solve the task correctly the sender is required to generate a referring expressions about the target object through the discrete channel while the receiver needs to resolve it. The experiment is set up in a way that avoids explicit human language information as e.g. human referring expressions or one-hot encoded attributes. Messages by the sender can only be based on the highly controlled implicit bias in the visual scenes.

Figure 2 shows the simplified architecture of the language game. The sender is given a set of

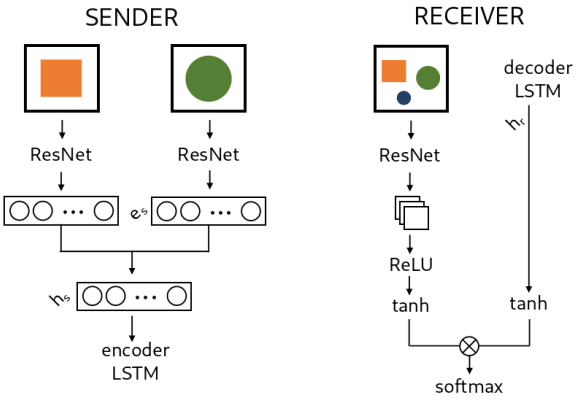


Figure 2: Simplified architecture of the attention predictor game.

bounding boxes of all objects in the scene, where the target object is always the first bounding box and the distractors are shuffled. The features of each bounding box are extracted using ResNet-101, combined and passed to the LSTM to produce a message. The receiver is shown the whole scene including the spatial information. Given the sender's message, the task is to predict the region around target object. For this, the image is divided into 14×14 regions. The target area is located around the center of the target object, consisting of 3×3 regions. The model is then tasked to predict the matrix $A = (a_{ij})$, where:

$$a_{ij} = \begin{cases} 1, & \text{if region } i,j \text{ in target area} \\ 0, & \text{otherwise} \end{cases}$$

The image is encoded using a combination of ResNet-101 and several convolutional layers described in Section 3.1. The resulting matrix has 128×14×14 dimensions, corresponding to the 14×14 regions.

The sender's message is decoded using an LSTM and the dot product is calculated between each encoded region of the image and the encoded message. The $softmax$ function is applied subsequently, which results in a 14×14 matrix. This emphasizes the correlation between the message and each region. A high dot product for a region indicates a high correlation between the message and the specific region, while a low dot product indicates the opposite. Training the agents like this should therefore highlight the regions in the image that are described by the sender, namely the region around the target object. To calculate the loss, the $softmax$ function is applied over the prediction and compared to the ground truth matrix

A using *binary cross entropy*. More details can be found in Appendix A. A total of 128.000 games are played. Furthermore, we allow message lengths of $n \in \{1, 2, 3, 4, 6\}$ and provide vocabulary sizes of $|V| \in \{2, 10, 16, 50, 100\}$.

The agents are evaluated on the summed predicted probability for the regions in the target area, the probability mass. In particular, the predicted matrix, consisting of probabilities for each region is multiplied with the ground truth matrix A, consisting of only ones and zeros. The result is summed and returns the probability mass for the target area. If the model predicts the target area perfectly, the probabilities in the target area sum to 1. If the model for instance focuses on the wrong object, the probability mass in the target area is lower.

All results are compared to a baseline in which the sender is generating random messages, so that the receiver needs to solve the task on its own. Any increase in performance requires information being transferred between the agents and the emergence of a language.

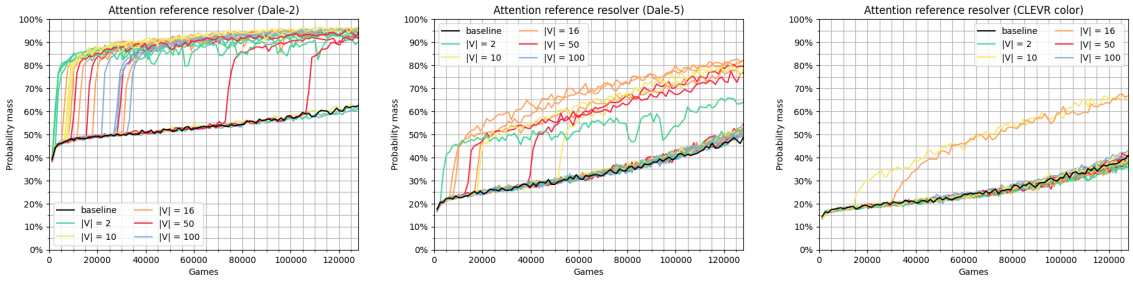
Results

The learning curves are shown in Figure 3. As can be seen, the agents are able to solve the task across all datasets, but with different consistency. However, when the agents start to learn to communicate, the probability mass is boosted instantly to a higher level, where it again learns at a slower speed parallel to the baseline. On the 'Dale-2' dataset, the boost is around 40% points. Most of the learning takes place in the first 40.000 games, but there are also two configurations that increase the performance very late after 70.000 and 105.000 games respectively. Hereby, agents tend to learn faster the smaller their vocabulary size is. Using the 'Dale-5' dataset, the probability masses are boosted around 30% points when the agents start to communicate successfully. Compared to the 'Dale-2' dataset, fewer configurations start to converge, while most achieve performances close to the baseline. The smaller number of learning curves makes the analysis more difficult, but the same trend about the vocabulary size is still visible. Interestingly, only one configuration with $|V| = 2$ beats the baseline, but behaves relatively unstable over the remaining training. On the other hand no configuration with $|V| = 100$ is successful. This indicates that one symbol is too few to encode all meaning, but too many symbols pose a too high difficulty to learn. This hypothesis is amplified by the results on the 'CLEVR color' dataset. Only two configurations beat the baseline, both with a medium-sized vocabulary size and message length. In both cases, the learning takes place relatively late, after 15.000 and 30.000 games respectively.

		Dale-2	**Dale-5**	**color**
n	$\|V\|$	P **mass**	P **mass**	P **mass**
baseline		62,16%	49,61%	41,68%
2	2	92,27%	52,15%	33,64%
3	2	94,52%	51,97%	37,09%
4	2	89,15%	51,98%	39,68%
6	2	59,68%	53,57%	38,43%
2	10	**96,16%**	80,26%	36,53%
3	10	94,9%	53,47%	38,24%
2	16	95,84%	**84,03%**	39,65%
4	10	**96,08%**	48,03%	**64,31%**
3	16	94,59%	**81,46%**	**67,88%**
6	10	63,46%	**82,12%**	40,11%
4	16	94,14%	49,81%	40,84%
6	16	95,86%	50,71%	40,61%
2	50	93,78%	52,24%	39,56%
3	50	93,88%	79,65%	40,36%
2	100	92,43%	53,23%	37,68%
4	50	**96,24%**	48,79%	43,61%
3	100	95,25%	48,52%	42,55%
6	50	91,27%	52,55%	40,21%
4	100	95,55%	49,65%	42,85%
6	100	60,27%	46,92%	41,98%

Table 1: Probability masses of the attention reference resolver after 128.000 games: n are different maximum message lengths and $|V|$ are different vocabulary sizes. Results in red didn't pass the baseline. The results are sorted by the product of n and $|V|$ which corresponds to available space for the message. The best results are achieved with a medium-sized message space across all datasets.

The final probability masses after 128.000 games are summed up in Table 1. Interestingly, the baseline can already find and attend to the correct regions in many cases without the help of the sender. The probability mass is higher than a uniform distribution ($\approx$ 4,6%) and a random guess of an object. It reaches 62,16% on the 'Dale-2' dataset, 49,61% on the 'Dale-5' dataset and 41,68% on the 'CLEVR color' dataset. Looking at the 'Dale-2' dataset, almost all configurations beat the baseline and achieve performances of over 90%, the best configurations reach even 96%. Only three configurations stay on the level of the baseline. When comparing the results, mostly the message length n seems to have an influence on the performance. While configurations with $n = 6$ can perform well, this is not constant. All three configurations that don't pass the baseline are allowed to produce message with $n = 6$. $n \in \{3, 4\}$ seem to help the



(a) 'Dale-2' dataset with different $|V|$ highlighted (b) 'Dale-5' dataset with different $|V|$ highlighted (c) 'CLEVR color' dataset with different $|V|$ highlighted

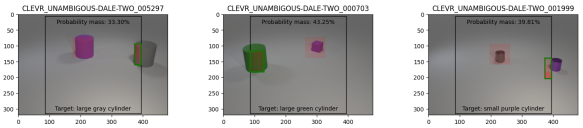
Figure 3: Learning curves of all language games on each dataset. The colors correspond to different vocabulary sizes $|V|$. The baseline is marked in black.

agents the most to perform consistently well, but the difference to configurations with $n = 2$ is very small. In both cases, the target object is unambiguously identified. The small number of experiments doesn't allow definite conclusions on the influence of the vocabulary size $|V|$, though $|V| = 2$ performs slightly worse than the remaining vocabulary sizes. A correlation between n and $|V|$ is not identifiable.

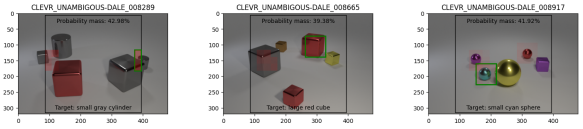
On the 'Dale-5' dataset, the agents already have bigger problems to beat the baseline. Only 8 out of 30 configuration perform better and reach probability masses around 76% to 84%. However, the increase compared to the baseline is as high as on the 'Dale-2' dataset, with around 30% points. Smaller message lengths ($n \in \{2, 3\}$) as well as a medium-sized vocabulary ($|V| \in \{10, 16, 50\}$) tend to help the agents more, to solve the task successfully. As before, no correlation is visible with the few successful games. That the agents struggle more with the 'Dale-5' dataset is not surprising. First, the larger number of distractors makes it more difficult for the receiver to focus, as can be seen already in the baseline performances. Additionally, the larger number of objects also influences the referring expression needed to uniquely describe the target object. With an increasing number of distractors, the probability rises that the target object shares attributes with any distractor. Therefore, it is more likely that the sender needs to use two or three attributes to describe the target object on the 'Dale-5' dataset compared to the 'Dale-2' dataset. This is naturally more complex to learn for the agents. Finally, since more objects are present, they are more likely clustered closer together, which can result in the identification of adjacent regions to the target regions.

The agents struggle the most on the 'CLEVR color' dataset. In this case, only two configurations perform better than the baseline and reach a probability mass of around 64% to 67%. Both utilize a medium message length of $n \in \{3, 4\}$ and a medium-sized vocabulary of $|V| \in \{10, 16\}$. Interestingly, several configurations with short message lengths of $n = 2$ perform worse than the baseline. This indicates that there is communication between the agents, but it rather distracts the receiver from the target object towards the distractors. The same point for a more difficult task when more objects are involved can be made for the 'CLEVR color' dataset. This dataset includes even up to 10 objects present in the scene which increases the likelihood that the receiver focuses on a wrong object.

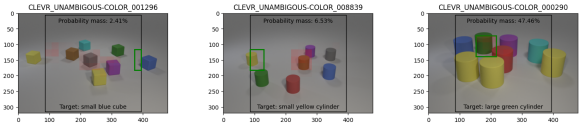
Figure 4 shows examples of the wrongly identified regions on each dataset. These are predictions by the agents that are wrong even though a language emerged successfully. Main problems seemed to be target objects not being in the actual frame of the scene that the receiver was processing. This happens due to center cropping the image to prepare it as input for the ResNet model. However, in several cases (as in the central image), especially for the 'Dale-5' dataset, all objects are visible, and the agents still don't attend solely on the target object. Rather than choosing one of the distractors, the agents usually attend to both objects relatively equally. This indicates that the receiver is uncertain which object the sender is describing. In contrast, the share of errors of the latter type is drastically higher. While a general pattern is difficult to identify, the receiver tends to confuse the target object with distractors that share multiple attributes with each other. In the central and right image, the wrongly identified distractors share both *size* and



(a) 'Dale-2'



(b) 'Dale-5'



(c) 'CLEVR color'

Figure 4: Examples of the predictions in language games with successful communication with a probability mass lower than 50% on the 'Dale' and 'CLEVR color' datasets. The black rectangle shows the cropped section the model is actually seeing after the image is preprocessed for ResNet-101. The green rectangle surrounds the target region that needs to be predicted while the red regions show the actual predictions of the model. The more intense the red, the higher is the probability that the model assigned to this region.

shape with the target object.

4.2 Sender and receiver with natural language referring expressions

Two further experiments are conducted to have a closer look at the sender and receiver models. More precisely, we evaluate how introducing explicit natural language bias into the models changes the training and ability to solve the tasks. This is done by training both the sender and receiver models separately outside a language game context, while the architecture stays the same. Instead of generating and respectively understanding a message of an emergent language, natural language referring expressions are used. Here, we know that the symbols are grounded in the visual scenes and correspond to attributes of the objects.

4.2.1 Referring expression generation (sender)

First, instead of generating a message for the receiver, the sender is now tasked to generate natural language referring expressions. The referring expressions for the target object are generated using the incremental GRE-algorithm (Dale and Reiter, 1995). By this, the model needs to describe the target object with respect to the distractor objects.

During testing, the LSTM is always forced to generate three tokens, with an embedded <sos> token as first input to the LSTM. Each token in the sequence is determined greedily, by selecting the highest logit in the output of each step in the LSTM. Training is done for 30 epochs and with a learning rate of 2×10^{-4}. The loss is calculated using cross entropy.

This task can be interpreted as a classification task rather than a natural language generation task, as the model is tasked to assign specific attributes to the target object instead of producing free text with a large vocabulary. Furthermore, the model's success is validated on accuracy, recall and precision scores. The **overall accuracy** is a measure if the model predicted every word in the referring expression correctly.

	Accuracy	**F1-Score**
Dale-2	99%	98,57%
Dale-5	69%	89,53%
CLEVR color	93%	95,17%

Table 2: Overall accuracies (Accuracy) and F1-Scores after 30 epochs with embedding size $e = 100$, $LSTM_o = 500$ and $LSTM_e = 30$.

Table 2 shows the *overall accuracy* and *F1 scores* for each word. As can be seen, the overall accuracies, in other words perfect matches of the generated referring expression depend very much on the dataset. With the 'Dale-2' and 'CLEVR color' dataset, the model can achieve high scores of 99% and 93% if the samples. In contrast, the model can only generate perfect referring expressions in 69% of the samples of the 'Dale-5' dataset.

Tables 3 and 4 give a more detailed insight in the results and especially what mistakes the model is making for both the 'Dale-5' and 'CLEVR color' datasets. The tokens are grouped by attribute and also show the metrics averaged over each of the attributes. The metrics of the <pad> token indicate if the model produced the correct length of the referring expression, in other words if it was able to determine which attributes are necessary to discriminate the target object from the distractors. For the 'CLEVR color' dataset, the scores are perfect. This is not surprising, since all referring expressions for the 'CLEVR color' dataset consist of exactly two attributes, shape and color, and the

		small	large	**size**	cube	cylinder	sphere	**shape**	<pad>
Dale-2	Precision	99,17	98,29	*98,73*	99,86	99,71	99,67	*99,75*	99,64
	Recall	97,54	94,26	*95,9*	100	99,56	99,67	*99,74*	99,77
Dale-5	Precision	69,65	69,21	*69,43*	98,19	98,32	98,39	*98,3*	82,22
	Recall	62,11	66,15	*64,13*	98,79	97,87	98,25	*98,3*	84,59
CLEVR color	Precision	-	-	**-**	100	100	100	*100*	100
	Recall	-	-	-	100	100	100	*100*	100

Table 3: Precision and Recall in % for <pad>, size and shape tokens with $e = 100$, $LSTM_o = 500$ and $LSTM_e = 30$. The columns **shape** and **size** show the average across all tokens of the respective attribute.

		blue	brown	cyan	gray	green	purple	red	yellow	**color**
Dale-2	Precision	94,51	98,77	97,59	98,68	98,89	98,8	97,47	100	*98,09*
	Recall	97,73	100	98,78	97,4	96,74	98,8	100	98,8	*98,53*
Dale-5	Precision	92,12	93,82	89,13	89,12	92,63	91,12	97,24	94,36	*92,44*
	Recall	92,12	89,78	94,91	94,51	95,71	92,42	89,34	94,85	*92,95*
CLEVR color	Precision	93,46	92,37	94,47	93,86	92,04	91,13	90,07	94,7	*92,76*
	Recall	92,75	92	95,98	89,92	94,12	91,13	94,23	91,91	*92,76*

Table 4: Precision and Recall in % for color tokens with $e = 100$, $LSTM_o = 500$ and $LSTM_e = 30$. The column **color** shows the average across all colors.

first generated token will always be the only <pad> token in the referring expression (corresponding to the unspecified size). The <pad> token is therefore easy to learn. For the 'Dale-5' dataset, the model struggles more to predict the correct length of the referring expression.

The shape can be identified very well across all datasets. The model predicts the correct shape for all samples using the 'CLEVR color' dataset, while both *precision* and *recall* lie around 98,3% when using the 'Dale-5' dataset. Even though the score is almost perfect, the slight difference might stem from the fact that all distractors have the same shape in the first case, while distractors can be different in the second case. Consequently, the model is only exposed to one shape at a time for each sample, which might simplify its identification.

For the color attribute, the metrics drop significantly for both 'Dale-5' and 'CLEVR color' to an average of around 93%. Hereby, no meaningful difference can be seen across the datasets, but there are differences between the colors. Some colors are predicted with *precision* and *recall* around 95% to 96%, while others are only around 90%. However, these differences are not reproducible across multiple runs and configurations. The best and worst predicted colors vary and no conclusions can be drawn which colors are easier to predict for the model.

Finally, the size is the most difficult attribute to predict for the model. Apart from the 'CLEVR color' dataset, where a size never needs to be predicted and also is never predicted, the metrics for the prediction of size tokens are the lowest across all tokens. They are the only mistakes, the model makes, when exposed to the 'Dale-2' dataset and the average *precision* lies around 23% below the average of predictions of the color for the 'Dale-5' dataset, while the average *recall* lies around 28,82% below. The reason why the *precision* is higher than the *recall* is the <pad> token, which is predicted very often instead of a token specifying the size. In fact, the opposite relationship is visible for the *precision* and *recall* for said token. The much higher absolute number of <pad> tokens leads to a smaller relative difference of %-points shown in the table. Again, no conclusion can be drawn if larger or smaller objects are easier to predict, since the results vary across runs and configurations.

In conclusion, the model successfully extracts discriminative features and produces referring expressions, though performance depends heavily on the number of distractors. Shape attributes are most easily identified, while size attributes prove most

challenging.

4.3 Referring expression resolution (receiver)

As before, the setup of the receiver model stays the same for this experiment, but instead of interpreting the sender's message, natural language referring expressions are passed to the model. As we know that the referring expressions are grounded in the scene, we can now compare the results to the language games, where the agents needed to learn and ground the arbitary vocabulary first.

	Probability mass
Dale-2	95,16%
Dale-5	92,19%
CLEVR color	95,33%

Table 5: Probability masses of the model after 20 epochs with $LSTM_e = 15$ and $LSTM_o = 1500$.

The results are shown in Table 5. Across all datasets, the model is able focus on the correct region in the image with high precision of over 90%. Interestingly, a different pattern emerges when comparing the results to the language games. While both agents and the single model achieved the best scores with the 'Dale-2' dataset, the single model can achieve similar results on the 'CLEVR color' dataset. On the 'Dale-5' dataset, the performance is slightly worse. In contrast, the agents achieved better results on the 'Dale-5' dataset, and struggled mostly with learning and grounding colors.

5 Discussion and Conclusion

We demonstrate a method for conducting focused experiments on artificial data through which we gain valuable insights what particular models are capable of learning from data and their dependence on the structure and representations in the data in the context of linguistic coordination and learning over a visual scene. This knowledge can be transferred to the design of larger systems that are trained on real data to gain insights about learning architectures, representations of features and datasets. They can also be used as a diagnostic probes for systems trained on real data.

Our language games revealed that agents can successfully develop communication protocols, achieving substantial performance gains over baselines. However, emergent communication faces constraints: medium-sized vocabularies and message lengths proved most effective. Scene complexity significantly impacts learning, with simpler scenes enabling near-perfect communication while complex scenes challenged most configurations.

The natural language experiments provided crucial insights into these limitations. When generating referring expressions, models achieved high accuracy on simple scenes but struggled with complex discriminations. Critically, the *size* attributes proved most difficult to learn across all tasks, followed by the *color*, while the *shape* was consistently well-identified. This indicates that humans and artificial neural networks have quite different learning biases that facilitate learning for humans (e.g. pragmatic referring described in the Dale-Reiter algorithm) is difficult to learn for systems. The experiments demonstrate that once we add such learning biases (e.g. modelling focused attention) learning becomes more successful. Overall, the results indicate that to be successful, learning language and vision models needs to go beyond mere observation of pixels and words.

Future work should investigate the linguistic properties of the emergent languages to better understand how agents encode visual attributes in their communicative protocols. Detailed analysis of message patterns could reveal whether emergent languages develop similar structures seen in natural languages.

References

Mark Bartlett and Dimitar Kazakov. 2005. The origins of syntax: from navigation to language. *Connection Science*, 17(3-4):271–288.

Herbert H. Clark. 1996. *Using language*. Cambridge University Press, Cambridge.

Robin Cooper. 2023. *From Perception to Communication: A Theory of Types for Action and Meaning*, volume 16 of *Oxford Studies in Semantics and Pragmatics*. Oxford University Press Press.

Robert Dale and Ehud Reiter. 1995. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive science*, 19(2):233–263.

Stevan Harnad. 1990. The symbol grounding problem. *Physica D 42: 335-346*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017a. 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. arXiv.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017b. Inferring and executing programs for visual reasoning. In *Proceedings of the IEEE international conference on computer vision*, pages 2989–2998.

Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. 2019. EGG: a toolkit for research on emergence of lanGuage in games. 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 55–60, Hong Kong, China. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Simon Kirby, Hannah Cornish, and Kenny Smith. 2008. Cumulative cultural evolution in the laboratory: An experimental approach to the origins of structure in human language. *Proceedings of the National Academy of Sciences*, 105(31):10681–10686.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2017. Multi-agent cooperation and the emergence of (natural) language. In *International Conference on Learning Representations*.

Terry Regier. 1996. *The human semantic potential: spatial language and constrained connectionism*. MIT Press, Cambridge, Massachusetts, London, England.

Deb Roy. 2005. Semiotic schemas: a framework for grounding language in action and perception. *Artificial Intelligence*, 167(1-2):170–205.

Luc Steels and Martin Loetzsch. 2009. Perspective alignment in spatial language. In Kenny R. Coventry, Thora Tenbrink, and John A. Bateman, editors, *Spatial Language and Dialogue*, volume 3 of *Explorations in language and space*, pages 70–88. Oxford University Press.

A Technical details of the language games

The sender extracts the features of each bounding box using ResNet-101 and projects them to an image embedding dimension $e_r = 100$ with a linear layer. All encoded bounding boxes are concatenated and again compressed to the decoder output dimension $h_s = 500$ using another linear layer. This representation of all objects serves as the initial hidden state of an LSTM, which generates the referring expression. Tokens used in the LSTM are embedded with embedding dimension $LSTM_{s,e} = 100$. During training, teacher forcing is applied by using embeddings of the ground truth tokens as the input sequence for the LSTM, instead of the output of the LSTM.

The receiver decodes the sender's message using an LSTM with a hidden size $h_r = 500$ and token embedding dimension of $LSTM_{r,e} = 100$. The image is encoded using a combination of ResNet-101 and several convolutional layers described in Section 3.1. Both encodings are passed through a $tanh$ non-linearity, and the results are combined using a dot product. The resulting vector is passed through a $softmax$ function to produce a probability distribution over the 14×14 regions of the image.

The experiments are conducted with the following hyperparameters: a learning rate of 2×10^{-4}, a temperature for the Gumbel-Softmax relaxation of 1 and *Adam* (Kingma and Ba, 2015) as optimizer.

The source code for all experiments is available at github.com/DominikKuenkele/MLT_Master-Thesis.

On the Role of Linguistic Features in LLM Performance on Theory of Mind Tasks

Ekaterina Kozachenko*
University of Lorraine
ETH Zürich

Gonçalo Guiomar
ETH Zürich

Karolina Stańczak
ETH Zürich

{ekozachenko, gguiomar, kstancza}@ethz.ch

Abstract

Theory of Mind presents a fundamental challenge for Large Language Models (LLMs), revealing gaps in processing intensional contexts where beliefs diverge from reality. We analyze six LLMs across 2,860 annotated stories, measuring factors such as idea density, mental state verb distribution, and perspectival complexity markers. Notably, and in contrast to humans, we find that LLMs show positive correlations with linguistic complexity. In fact, they achieve high accuracy (74-95%) on *high* complexity stories with explicit mental state scaffolding, yet struggle with *low* complexity tasks requiring implicit reasoning (51-77%). Furthermore, we find that linguistic markers systematically influence performance, with contrast markers decreasing accuracy by 5-9% and knowledge verbs increasing it by 4-10%. This inverse relationship between linguistic complexity and performance, contrary to human cognition, may suggest that current LLMs rely on surface-level linguistic cues rather than genuine mental state reasoning.

1 Introduction

While Large Language Models (LLMs) demonstrate remarkable capabilities in code generation (Jiang et al., 2024), multilingual translation (Zhu et al., 2024), and long-context conversational memory (Liu et al., 2024), their performance on social reasoning tasks remains fundamentally unreliable. Although LLMs are approaching human accuracy on simple false-belief tests (Moghaddam and Honey, 2023; Kosinski, 2024), their inconsistent patterns on more sophisticated tasks requiring social reasoning (Sap et al., 2022; Kim et al., 2023), suggest they rely on mechanisms fundamentally different from human cognition.

At the heart of this reasoning lies Theory of Mind (ToM), the human ability to model others' mental states, especially when their beliefs contradict reality (Premack and Woodruff, 1978). Classic false-belief tasks, such as the Sally-Anne test, probe this ability by requiring a model to predict an agent's actions based on their incorrect beliefs. Computationally, this requires processing intensional contexts created by attitude verbs like "believe," where the truth of a proposition is evaluated relative to a subjective perspective rather than objective reality (Montague, 2008). Recent findings reveal that LLMs capable of passing standard false-belief tests often fail on their minor variations (Ullman, 2023). This suggests they lack a robust understanding of how mental state verbs create distinct semantic contexts that block standard entailment (Karttunen, 1973).

In this paper, we empirically analyze six LLMs on ToM tasks to understand their failure patterns on tasks requiring semantic reasoning. We examine 2,860 stories by quantifying linguistic features related to information structure (idea density) and lexical patterns (mental state verb density). We also manually annotate each story for its level of perspectival complexity and linguistic markers. We address three key research questions: (**RQ1**) To what extent do idea density and mental state verb density correlate with LLM performance on mental state reasoning? (**RQ2**) How do linguistic markers of perspectival complexity influence model performance on ToM tasks? (**RQ3**) What systematic failures emerge across different model architectures?

We find that LLMs exhibit opposite correlations to humans in terms of linguistic complexity, yet paradoxically achieve the highest accuracy on high-complexity stories with explicit mental state scaffolding. These findings suggest that LLMs rely on surface linguistic cues rather than genuine perspective-tracking.

*This research was conducted while visiting ETH Zürich.

Proceedings of the 16th International Conference on Computational Semantics, pages 308–316
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

2 A Semantic Framework for Theory of Mind

To formally analyze ToM, the capacity to attribute beliefs, desires, and intentions to oneself and others, and to recognize that these states may diverge from reality (Premack and Woodruff, 1978; Astington, 1993), we ground our analysis in a multi-agent epistemic–doxastic logic (Hintikka, 2005; Fagin and Halpern, 1994). This framework provides a precise language for representing nested perspectives and allows for systematic categorization of the perspectival complexity of social reasoning scenarios (Karttunen, 1971; Giannakidou, 1998).

A mental-state representation in our framework consists of an agent $a \in \mathcal{A}$ from a set of story participants, an attitude (e.g., knowledge K, belief B), and a content formula φ expressing a proposition about events or states. Our formal language $\mathcal{L}_{[1][2]...[n]}$ extends propositional logic $\mathcal{L}$ with a set of modal operators $[i]$, each corresponding to a mental attitude held by a specific agent. For instance, for agents $a, b \in \mathcal{A}$, the formula $K_a\varphi$ expresses "a knows φ," and $B_b\varphi$ expresses "b believes φ." These operators can be nested to represent higher-order ToM, as in $K_aB_b\neg p$ ("a knows that b believes that p is false").

The semantics are defined using a generalized Kripke model (Voorbraak, 1992), a tuple

$$M = \langle w_0, W, \{\Sigma_1, \Sigma_2, \ldots, \Sigma_m\}, \langle \sigma_1, \sigma_2, \ldots, \sigma_m\rangle, \langle F_1, F_2, \ldots, F_m\rangle, \vDash\rangle$$

where W is a set of possible worlds, Σ_i is a non-empty set of epistemic states for attitude i, $\sigma_i : W \to \Sigma_i$ maps each world to an epistemic state, F_i is a set of projection functions that extract information from an epistemic state, and $\vDash$ is the valuation function, where $\vDash (w, [i]\varphi)$ depends on the epistemic state $\sigma_i(w)$. The key insight of the generalized Kripke models is that epistemic states are explicitly represented as atomic entities, not sets of worlds, with nonstandard valuation for modal operations.

We instantiate this general framework for two attitudes: objective knowledge and rational belief.

Objective Knowledge. Modeled as an S5 modality, objective knowledge corresponds to truthful, introspective information. In an *objective knowledge (OK) model*, the truth condition for $K_a\varphi$ is given as:

$$w \vDash K_a\varphi \text{ iff } \forall w' \in W\big(\kappa(w') = \kappa(w) \Rightarrow w' \vDash \varphi\big)$$

where $\kappa(w)$ is the information state at world w. This states that φ is true in all worlds that are informationally indistinguishable from w.

Rational Belief. Modeled as a KD45 modality, rational belief is not necessarily true but is consistent and introspective. In a *rational belief (RIB) model*, the belief set $\|\beta(w)\|_B$ for a state $\beta(w)$ is non-empty (*consistency*) and constant across all worlds within that set (*introspection*). The truth condition for $B_a\varphi$ is:

$$w \vDash B_a\varphi \quad \text{iff} \quad \forall w' \in \|\beta(w)\|_B \; w' \vDash \varphi.$$

This states that φ is true in all worlds compatible with the agent's beliefs.

Veridicality. Following Karttunen (1971, 1973) and Giannakidou (1998), we classify attitude verbs by their entailment properties. An operator is *veridical* if it entails its complement φ in the actual world (e.g., "know", "realize"), *non-veridical* if it carries no such entailment (e.g., "believe", "suspect"), and *anti-veridical* if it entails $\neg\varphi$ (e.g., "pretend", "imagine"). As a subset of non-veridical operators (Giannakidou, 2013), an operator F is anti-veridical if $F\varphi$ is false in an agent's epistemic model $M(x)$, i.e., $M(x) \cap [\![\varphi]\!] = \varnothing$. We note that this distinction can be modeled within the non-veridical *RIB* framework by adding a constraint that all accessible worlds satisfy $\neg\varphi$ (e.g., for attitudes like "pretend" or "imagine"); however, our analysis focuses on the core attitudes of knowledge and belief.

Perspectival Complexity. We quantify complexity based on the nesting depth of modal operators and the number of distinct agents. Depth 0 (no operators) is *simple*. Depth 1 with a single agent is *low* complexity. Depths 2 with multiple agents are *medium*, and depths of 3+ with at least three agents are *high*. We also annotate linguistic markers, including explicit contrasts ($B_a\varphi \wedge \neg\varphi$) and displacement (a proposition φ appearing only within the scope of an operator).

3 Methodology

3.1 Data

For our analysis, we use the English portion of ToMBench (Chen et al., 2024), a benchmark

designed to assess ToM capabilities in LLMs. ToMBench covers 31 distinct aspects of social cognition organized into six categories: *beliefs* (reasoning about divergent or false mental states), *emotions* (understanding situational feelings), *intentions* (recognizing goal-directed actions), *knowledge* (tracking access to information), *non-literal communication* (interpreting indirect meaning), and *desire* (identifying subjective wants). Representative examples from the dataset are provided in the App. A in Tab. 1. Every instance of the ToMBench contains a story, followed by a question, and four plausible options (A, B, C, D) where only one answer is correct and the others are high-quality but misleading wrong answers.

Data annotation. We manually annotated each instance in the dataset for two key properties: *perspectival complexity* and the presence of specific *linguistic markers*. We categorized stories into four levels based on mental state attribution patterns: *simple story* (no explicit mental state attributions), *low* (single agent with mental state), *medium* (multiple agents or belief-reality contrasts), and *high* (nested mental states or three+ agents with contrastive structures). We tracked three types of linguistic markers: (1) contrast markers signaling belief-reality divergence ("but actually," "however"), (2) displacement markers indicating perspective shifts ("from X's perspective"), and (3) verb types distinguishing factive (knows, sees) from non-factive (thinks, believes) mental states. While this surface-level annotation simplifies true intensional complexity, which would require analyzing scope ambiguities, de re/de dicto distinctions, and semantic properties of embedded clauses, it captures identifiable correlates that may proxy for deeper semantic complexity. This approach tests whether LLMs are sensitive to surface markers of perspective complexity, even if we cannot directly assess their handling of formal intensional semantics.

The annotation was performed by a linguistics expert and validated by a second expert, both authors of this work. All discrepancies were resolved through discussion, resulting in a high inter-annotator agreement (Cohen's $\kappa = 0.90$ for complexity and $\kappa = 0.95$ for markers). A detailed guide to our annotation criteria is available in App. A. Additionally, we automatically computed Idea Density and lexical patterns via Mean Syntactic Verb Dependency for each instance using

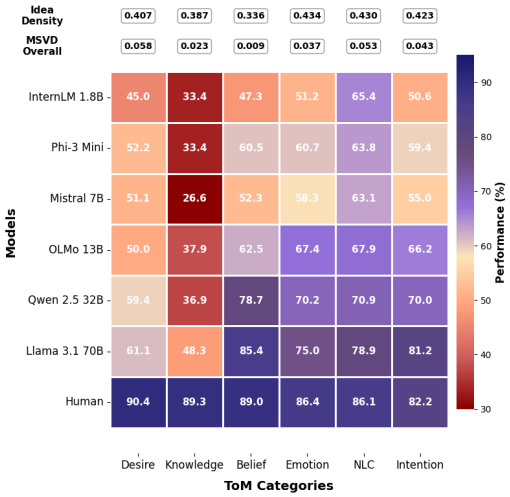


Figure 1: Heatmap of performance (%) for LLMs and humans across six ToM categories with the average idea density and MSVD for stories in each category.

custom scripts based on spaCy.[1]

Idea Density (ID). Idea density measures the rate of elementary propositions in a text, normalized by its length. It serves as a metric for informational complexity, where lower density has been linked to cognitive decline and an increased risk of Alzheimer's disease (Sirts et al., 2017). The idea density for a given text is calculated as:

$$\text{Idea Density} = \frac{\text{Number of Propositions}}{\text{Number of Words}} \quad (1)$$

Mean Syntactic Verb Dependency (MSVD). To capture the expression of characters' internal states, which is a key component of ToM (Astington, 1993), we measure the density of state verbs. State verbs (*e.g.*, *think, know, believe, want, feel*) describe cognitive or emotional states rather than physical actions. A higher frequency of these verbs can indicate a greater focus on intentionality and mental representation within a story. For a story S, we calculate MSVD as:

$$MSVD(S) = \frac{|V_{\text{state}}(S)|}{N_{\text{words}}(S)} \quad (2)$$

where $V_{\text{state}}(S)$ is the set of lemmatized state verbs in the text and $N_{\text{words}}(S)$ is the total word count.

3.2 Models

We evaluate several state-of-the-art LLMs on the ToMBench benchmark, ranging from 1.8B to 70B

[1] https://spacy.io/

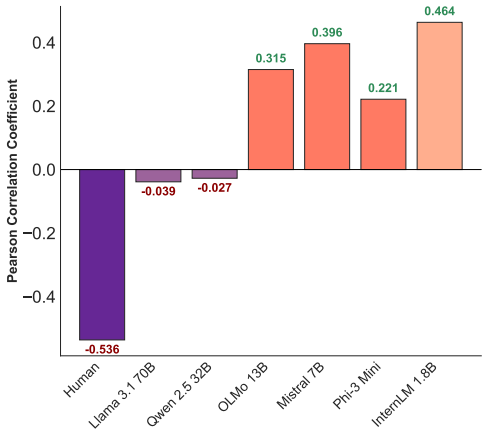


Figure 2: Pearson correlation between idea density and task performance. A strong negative correlation is observed for humans, in contrast to most models.

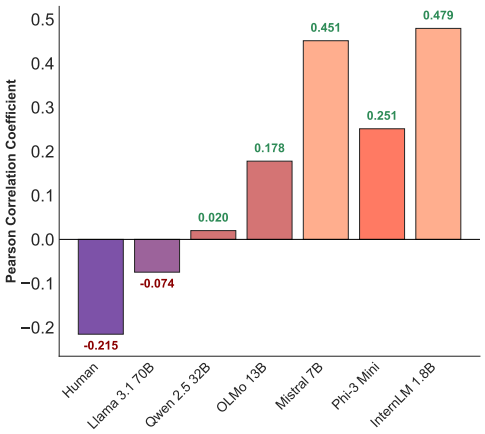


Figure 3: Pearson correlation between MSVD and task performance for humans and LLMs. A negative correlation is observed for humans ($r = -0.215$), while most models exhibit a positive correlation.

parameter count: LLama-3.1-70B (Touvron et al., 2023), Qwen-2.5-32B (Team, 2025), OLMo-2-13B (OLMo et al., 2024), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), Phi-3-Mini-4k-Instruct (Abdin et al., 2024), and InternLM-2.5-1.8 (Cai et al., 2024). To this end, we prompt these models to answer the tasks from the dataset discussed above in the multiple-choice setup.

4 Experiments and Results

RQ1: To what extent do idea density and mental state verb density correlate with LLM performance on mental state reasoning? We first examine performance across the six ToM categories shown in Fig. 1, revealing a consistent human advantage across all categories. To investigate the relationship between linguistic features and success on mental state reasoning tasks, we analyze the correlation between performance on ToMBench and two textual features: idea density and MSVD. We compute the Pearson correlation between these features and task performance across both the human baseline and the suite of evaluated LLMs. The human performance data is derived from the original study involving 20 graduate students (Chen et al., 2024). Our analysis reveals a stark, opposing relationship between these linguistic features and performance for humans versus LLMs. In Fig. 2, we observe a negative correlation for human performance with both ID ($r = -0.536$) and MSVD ($r = -0.215$). This indicates that as texts become more informationally dense or contain more explicit mental state verbs, human performance on the ToM tasks tends to decrease. In direct contrast,

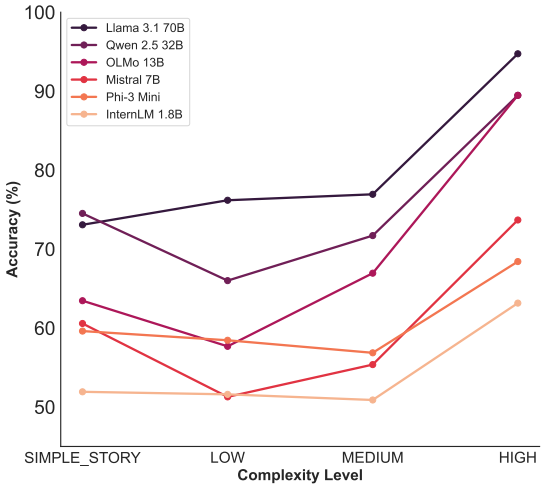


Figure 4: LLMs performance across perspectival complexity categories.

LLMs consistently show a positive correlation with these same features. The correlation between performance and ID is positive across most models, ranging to a moderate $r = 0.464$. A similar positive trend is observed for MSVD (see Fig. 3). This suggests that, unlike humans, LLM performance and comprehension are enhanced through increased linguistic scaffolding.

RQ2: How do linguistic markers of perspectival complexity influence model performance on ToM tasks To investigate the impact of narrative structure, we evaluated LLM accuracy across four levels of perspectival complexity (Fig. 4). Our results reveal a "complexity paradox:" contrary to expectations, models achieve peak performance (74-95% accuracy) on *high* complexity stories with

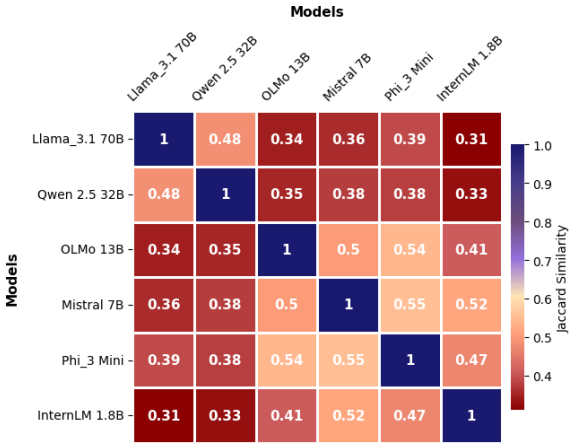


Figure 5: A Jaccard similarity matrix illustrating the degree of overlap in errors between model pairs, where higher values signify more similar failure modes.

nested mental states, while struggling most with the *low* complexity category (51-77%). This suggests that explicitly complex narrative structures may provide a form of linguistic scaffolding that aids model reasoning more than the subtler challenges of medium-complexity texts.

RQ3: What systematic failures emerge across different model architectures? To identify systematic failures across architectures, we computed the Jaccard similarity of incorrect responses for all model pairs, as shown in Fig. 5. The results reveal a clear cluster of smaller models (Mistral 7B, Phi-3 Mini, OLMo 13B) that exhibit high error overlap ($J \approx 0.5 - 0.55$), suggesting they share a common failure mode. In contrast, the largest models show more idiosyncratic errors, indicating they may overcome some specific systematic challenges. Moreover, we identified 245 stories (8.6%), where all models fail universally, concentrated in *low* (9.7%) and *medium* (6.9%) complexity levels. These systematic failures occur despite the presence of linguistic markers: stories with contrast markers, knowledge verbs, or moderate MSVD still cause universal failure when they require reasoning beyond surface cues. This pattern reinforces our finding that LLMs rely on explicit linguistic scaffolding: they fail systematically when answering correctly requires inference rather than pattern-matching on mental state markers.

This aligns with Ross and Pavlick (2019), who showed NLI models like BERT fail on non-veridical verbs (e.g., "think", "believe") due to pattern-matching biases rather than true inference. In our universal failure cases, similar non-veridical mental state verbs dominate low-complexity stories requiring implicit reasoning, while veridical "knowledge verbs" provide insufficient scaffolding, extending their veridicality bias to ToM contexts.

5 Related Work

Early ToM evaluations revealed superficial success on classic false-belief tasks, such as the Sally-Anne test (van Duijn et al., 2023), prompting more rigorous benchmarks. Recent work like ToMBench (Chen et al., 2024) and EPITOME (Jones et al., 2024) benchmarks show a recurring pattern of models handling basic belief-tracking but failing on tasks requiring pragmatic or social inference.

This weakness in compositional reasoning, also probed by procedurally generated narratives in ExploreToM (Sclar et al., 2025), suggests models exploit statistical shortcuts rather than genuinely tracking mental states. Other work reveals failures in more fundamental capabilities, such as the Two Word Test study (Riccardi and Desai, 2023). A common finding across these methods is that models often succeed by exploiting statistical patterns rather than by genuinely tracking mental states. However, prior work has not systematically distinguished between tasks with low and high intentionality (*i.e.*, simple belief attribution versus complex deception) or investigated how specific linguistic features influence LLM performance on ToM tasks. Our work aims to address these gaps.

6 Conclusions

We analyzed linguistic features in LLM performance on ToM tasks, revealing surprising patterns: (1) LLMs show positive correlations with idea density and MSVD, opposite to humans' negative correlations, (2) Models paradoxically excel on *high* complexity stories (74-95%) while struggling with *low* complexity (51-77%), and (3) All models fail systematically when implicit reasoning is required. These patterns suggest LLMs may leverage explicit linguistic markers rather than genuine mental state reasoning, though our correlational analysis cannot prove causation. The complexity paradox, where explicit mental state scaffolding aids performance, warrants further causal investigation to understand whether models truly rely on surface cues or develop deeper representations.

7 Limitations

While this study provides novel insights into the relationship between linguistic features and LLM performance on ToM tasks, we acknowledge several limitations that frame avenues for future research. First, our primary metrics, Idea Density and MSVD, are by design surface-level proxies for informational and perspectival complexity. While effective for establishing high-level correlation, these features do not capture the fine-grained syntactic and semantic structures that underpin intensional reasoning. Future work should augment this analysis with more structurally aware features. Second, our four-level classification of perspectival complexity may simplify a multifaceted phenomenon into discrete categories. However, this operationalization was necessary to analyze performance trends. A more fine-grained, continuous complexity score could enable a more nuanced regression analysis in future studies. Finally, our conclusion that LLMs rely on "linguistic scaffolding" and heuristics is drawn from the observed performance patterns and correlations. This study demonstrates that models behave in a way consistent with heuristic-based processing, but does not isolate the precise nature of these heuristics. A crucial next step, is to move from correlation to causation.

Acknowledgments

Gonçalo Guiomar and Karolina Stańczak were supported by ETH AI Center postdoctoral fellowships.

References

Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Hassan Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahmoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Xihui (Eric) Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Olatunji Ruwase, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Weijian Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone. Technical Report MSR-TR-2024-12, Microsoft.

Janet W. Astington. 1993. *The Child's Discovery of the Mind*. The Developing Child. Harvard University Press, Cambridge, MA.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, Chao Xu, Ruiliang Xu, Hang Yan, Yirong Yan, Xiaogui Yang, Haochen Ye, Huaiyuan Ying, Jia Yu, Jing Yu, Yuhang Zang, Chuyu Zhang, Li Zhang, Pan Zhang, Peng Zhang, Ruijie Zhang, Shuo Zhang, Songyang Zhang, Wenjian Zhang, Wenwei Zhang, Xingcheng Zhang, Xinyue Zhang, Hui Zhao, Qian Zhao, Xiaomeng Zhao, Fengzhe Zhou, Zaida Zhou, Jingming Zhuo, Yicheng Zou, Xipeng Qiu, Yu Qiao, and Dahua Lin. 2024. InternLM2 technical report. *arXiv preprint arXiv:2403.17297*.

Zhuang Chen, Jincenzi Wu, Jinfeng Zhou, Bosi Wen, Guanqun Bi, Gongyao Jiang, Yaru Cao, Mengting Hu, Yunghwei Lai, Zexuan Xiong, and Minlie Huang. 2024. ToMBench: Benchmarking theory of mind in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15959–15983, Bangkok, Thailand. Association for Computational Linguistics.

Max van Duijn, Bram van Dijk, Tom Kouwenhoven, Werner de Valk, Marco Spruit, and Peter van der Putten. 2023. Theory of mind in large language models: Examining performance of 11 state-of-the-art models vs. children aged 7-10 on advanced tests. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 389–402, Singapore. Association for Computational Linguistics.

Ronald Fagin and Joseph Y. Halpern. 1994. Reasoning about knowledge and probability. *Journal of the ACM*, 41(2):340–367.

Anastasia Giannakidou. 1998. *Polarity Sensitivity as (Non)Veridical Dependency*. Linguistik Aktuell/Linguistics Today. John Benjamins Publishing Company.

Anastasia Giannakidou. 2013. (non)veridicality, evaluation, and event actualization: evidence from the subjunctive in relative clauses. In Maite Taboada and Ljiljana Tvranc, editors, *Nonveridicality, Perspective, and Discourse Coherence*, Studies in Pragmatics Series. Brill.

Jaakko Hintikka. 2005. *Knowledge and Belief: An Introduction to the Logic of the Two Notions*. Texts in Philosophy. King's College Publications.

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. *arXiv preprint arXiv:2310.06825*.

Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. 2024. A survey on large language models for code generation. *arXiv preprint arXiv:2406.00515*.

Cameron R. Jones, Sean Trott, and Benjamin Bergen. 2024. Comparing humans and large language models on an experimental protocol inventory for theory of mind evaluation (EPITOME). *Transactions of the Association for Computational Linguistics*, 12:803–819.

Lauri Karttunen. 1971. Some observations on factivity. *Paper in Linguistics*, 4(1):55–69.

Lauri Karttunen. 1973. Presuppositions of compound sentences. *Linguistic Inquiry*, 4(2):167–193.

Hyunwoo Kim, Melanie Sclar, Xuhui Zhou, Ronan Bras, Gunhee Kim, Yejin Choi, and Maarten Sap. 2023. FANToM: A benchmark for stress-testing machine theory of mind in interactions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14397–14413, Singapore. Association for Computational Linguistics.

Michal Kosinski. 2024. Evaluating large language models in theory of mind tasks. *Proceedings of the National Academy of Sciences*, 121(45).

Na Liu, Liangyu Chen, Xiaoyu Tian, Wei Zou, Kaijiang Chen, and Ming Cui. 2024. From LLM to conversational agent: A memory enhanced architecture with fine-tuning of large language models. *arXiv preprint arXiv:2401.02777*.

Shima Rahimi Moghaddam and Christopher J. Honey. 2023. Boosting theory-of-mind performance in large language models via prompting. *arXiv preprint arXiv:2304.11490*.

Richard Montague. 2008. *The Proper Treatment of Quantification in Ordinary English*, volume 49. Springer, Dordrecht.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*.

David Premack and Guy Woodruff. 1978. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1(4):515–526.

Nicholas Riccardi and Rutvik H. Desai. 2023. The two word test: A semantic benchmark for large language models.

Alexis Ross and Ellie Pavlick. 2019. How well do nli models capture verb veridicality? 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 2230–2240.

Maarten Sap, Ronan Le Bras, Daniel Fried, and Yejin Choi. 2022. Neural theory-of-mind? on the limits of social intelligence in large LMs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3762–3780, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Melanie Sclar, Jane Dwivedi-Yu, Maryam Fazel-Zarandi, Yulia Tsvetkov, Yonatan Bisk, Yejin Choi, and Asli Celikyilmaz. 2025. Explore theory of mind: program-guided adversarial data generation for theory of mind reasoning. In *The Thirteenth International Conference on Learning Representations*.

Kairit Sirts, Olivier Piguet, and Mark Johnson. 2017. Idea density for predicting Alzheimer's disease from transcribed speech. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 322–332, Vancouver, Canada. Association for Computational Linguistics.

Qwen Team. 2025. Qwen2.5 technical report. *arXiv preprint arXiv: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 preprint arXiv:2302.13971*.

Tomer Ullman. 2023. Large language models fail on trivial alterations to theory-of-mind tasks. *arXiv preprint arXiv:2302.08399*.

Frans Voorbraak. 1992. Generalized Kripke models for epistemic logic. In *Proceedings of the Fourth Conference on Theoretical Aspects of Reasoning about Knowledge*, TARK '92, page 214–228, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. Multilingual machine translation with large language models: Empirical results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2765–2781, Mexico City, Mexico. Association for Computational Linguistics.

Story: Xiao Ming receives a bicycle on his birthday. **Ability:** Emotion **Question-1:** What is Xiao Ming's emotion? (A) Embarrassed **(B) Happy** (C) Disappointed (D) Regretful **Ability:** Belief **Question-2:** He should be very happy, but he is very disappointed, why? (A) Xiao Ming worries that riding a bicycle affects his studies. (B) Xiao Ming fears that riding a bicycle to school makes his classmates laugh at him. (C) Xiao Ming thinks the color of the bicycle does not match his clothes. **(D) Xiao Ming hopes for a computer as a gift, not a bicycle.** **Ability:** Emotion **Question-3:** Xiao Ming is having a birthday, he hopes for a computer or a new game as a birthday gift, on his birthday he receives a bicycle. What is Xiao Ming's emotion at this time? (A) Embarrassed (B) Happy **(C) Disappointed** (D) Regretful
Story: Almost every letter to Laura Company contains a check. Today, Laura receives 5 letters. Laura tells you on the phone "I look at 3 out of 5 letters. There are checks in 2 of the letters." **Ability:** Knowledge **Question-1:** Before Laura calls you, how many of these 5 letters do you think contain checks? (A) 0 (B) 1 (C) 2 **(D) 4** **Question-2:** After Laura calls you, how many of these 5 letters do you think contain checks? (A) 0 (B) 1 (C) 2 **(D) 4**

Table 1: Example of the theory of mind questions from the ToMBench.

A Additional Data Details

In Tab. 1, we show a few examples from the ToMBench that we use for the analysis.

Mapping Semantic Domains Across India's Social Media: Networks, Geography, and Social Factors

Gunjan Anand
University of Illinois
Urbana-Champaign,
Urbana, IL 61801 USA
gunjana2@illinois.edu

Jonathan Dunn
University of Illinois
Urbana-Champaign,
Urbana, IL 61801 USA
jedunn@illinois.edu

Abstract

This study examines socially-conditioned variation within semantic domains like kinship and weather using thirteen Indian cities as a case-study. Using bilingual social media data, we infer six semantic domains from corpora representing individual cities with a lexicon including terms from English, Hindi and Transliterated Hindi. The process of inferring semantic domains uses character-based embeddings to retrieve nearest neighbors and Jaccard similarity to operationalize the edge weights between lexical items within each domain. These representations reveal distinct regional variation across all six domains. We then examine the relationship between variation in semantic domains and external social factors such as literacy rates and local demographics. The results show that semantic domains exhibit systematic influences from sociolinguistic factors, a finding that has significant implications for the idea that semantic domains can be studied as abstractions distinct from specific speech communities.

1 Introduction

India is a country with many diverse cultures and languages. This creates interactions between languages, particularly Hindi and English in the Northern regions of India and between Hindi and other languages elsewhere. This paper asks whether such longstanding linguistic and cultural contact changes the character of semantic domains present within thirteen Indian cities. Much previous work views semantic domains as language-specific, so that a language like Hindi has a single semantic map for a domain like kinship. The contribution of this paper is to show that social factors like language contact have a systematic influence on the structure of semantic domains. India provides an ideal case-study because of these longstanding contact situations.

We focus on six semantic domains: weather, kinship, emotion, animals, professions and temporal units. These domains were chosen because of their known variation in lexical granularity between English and Hindi. For example, Hindi distinguishes between paternal and maternal grandfathers lexically, whereas English uses the same term for both relationships. Additional modifiers (*paternal, maternal*) are used in English when necessary. In contrast, Hindi uses the same term for *yesterday* and *tomorrow*, disambiguating based on verb tense and context; English uses distinct words for these two concepts. These examples show how languages can encode conceptual distinctions with differing levels of granularity. A speaker's lexical choices are shaped by grammatical and cultural systems enforced in the lexicon. Our question here is the degree to which linguistic and cultural contact create variation within semantic domains within the same languages.

To investigate this question, we analyze data from Indian social media. This kind of spontaneous, everyday language use provides insight into how different populations lexicalize these six semantic domains. Social media offers a large-scale, naturalistic corpus to capture regional variation. In particular, it allows us to ask whether the same semantic concepts are realized with consistent lexical patterns across cities or whether these patterns diverge due to differences in language contact and social environment. We develop a corpus of over 50 million samples containing a mix of English and Hindi usage across thirteen Indian cities as a means of observing semantic domains across regional populations.

Given population-specific corpora, we need to infer a representation of these semantic domains in order to compare them across populations. We take a data-driven approach based on non-contextual character embeddings from fastText, learning a separate model from each city-specific corpus. These embeddings can be seen as approximations of con-

Proceedings of the 16th International Conference on Computational Semantics, pages 317–330
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

ceptual structure in which lexical items from the same domain form a neighborhood within the embedding space. This approach to operationalizing a semantic domain as similarities within an embedding space aligns with an opposition theory approach to signs (de Saussure, [1916] 1983). This approach posits that the value of a concept is determined by its contrasting relations within the system of language, particularly how it contrasts with other similar terms. Therefore, embeddings offer a way to operationalize the structure of these semantic domains which can then be used to measure the degree to which these domains vary across speech communities.

Importantly, many concepts in these six domains exhibit co-lexification: there is not a one-to-one mapping between form and meaning. For example, the cases of *paternal/maternal grandfather* (in English) and of *yesterday/tomorrow* (in Hindi) are instances in which one language co-lexifies what the other splits into two separate items. In our bilingual corpus data, however, a speaker is not limited to the co-lexification patterns of either language. We hypothesize that this provides additional flexibility to the mapping between form and meaning within lexical items, allowing them to vary systematically across populations due to sociolinguistic conditions. If this is the case, we would expect that the operationalizations of these semantic domains, created using an embedding space, will also differ across regions in predictable ways.

This paper makes three main contributions: First, we show that these semantic domains, as inferred from corpora, vary significantly across Indian cities in way that corresponds with different levels of overall language contact. Second, we show that these variations are relatively stable across all six domains and are not artifacts within only a single domain. And, third, we show that these variations are not simply random but are significantly related to social and demographic factors. Taken together, these findings suggest that semantic domains are not a single entity shared by all speakers of a language but rather systems which are influenced by social factors like differing degrees of language contact.

After reviewing related work in semantic domains and social factors in Section 2, we present a dataset derived from Twitter/X posts from various Indian cities in Section 3. This dataset contains samples in English, Hindi and Transliterated Hindi. Our method for operationalizing semantic domains using an embedding space is detailed in Section 4, along with the social factors used for later analysis. The analysis of variation in semantic domains across cities is presented in Section 5, with a special focus on the relationship between these variations and external social factors like language contact. We end, in Section 6, by discussing the larger implications of this work on the interface between sociolinguistics and computational semantics. While previous computational work has abstracted away from sociolinguistic factors in the representation of semantic domains, the findings in this paper show that such idealized representations will not capture variations within the speech community.

2 Previous Work

2.1 Computational Approaches to Semantic Analysis

Word embeddings have become a widely used computational method for analyzing contextual relationships between words used in corpus data. Mikolov et al. (2013) suggests that embeddings allow semantic similarity to be mapped and quantified through vector proximity in embedding high dimensional spaces. This is further demonstrated studies such as in Jatnika et al. (2019) and Jin and Schuler (2015) which confirm that words which share similar contexts tend to cluster together in embedding spaces.

As explained by Lai et al. (2015), these models generate word vectors based on surrounding context, allowing semantic relatedness to be inferred by vector proximity. This aligns with opposition theory (de Saussure, [1916] 1983) as if a vector gets its value by the opposition vectors, semantic relatedness can be seen by how close the vectors are. So one would wonder how semantic domains can be seen in embedding spaces and how variant this would be within the domains?

Recent work has focused on applying this framework to study semantic domains. Grand et al. (2018) used embedding spaces to project out semantic domains (e.g. animals, weather, professions), showing how humans mentally organize semantic fields through patterns of usage. However, Antoniak and Mimno (2018) cautions the usage of such frameworks as these results may be sensitive to corpus size and sampling variability which raises concerns about how reproducible and conclusive the results can be. They suggest bootstrapping over multiple samples which is used to check stability of the model in this paper. Similarly, Burdick et al. (2021)

report variation in embedding stability across languages, particularly in morphologically rich contexts - an insight which is important to keep in mind while looking at India's multilingual landscape.

fastText, developed by Bojanowski et al. (2017) represents a significant advancement in creating embedding spaces for words in a corpus. The model has the ability to capture sub-word information which would help in analyzing morphologically rich languages like Hindi. Studies such as Rana et al. (2024) and Thavareesan and Mahesan (2020) have used fastText embeddings to analyze semantic similarity, confirming the model's strength in multilingual environments.

Building on these methods, this study extends prior work by analyzing semantic domain variation across cities in India. It uses fastText embeddings to map word usage onto high dimensional embedding spaces where each lexical item is represented as a vector.

Following Grand et al. (2018)'s framework, we construct semantic domain networks using embeddings. We use Jaccard similarity between k nearest neighbors of the lexical items to detect semantic similarity as supported by Gonen et al. (2020) as a stable and interpretable method for detecting semantic relationships. The study innovates by using these similarities to create a semantic domain structure to enable a more nuanced analysis of cross-regional variation. This use of embedding space to show dialectal and regional differences is seen in Dunn (2023), which demonstrates that the stability of embeddings vary significantly across geographically distinct corpora. This drives our city wise analysis of domain structures, allowing us to visualize how lexical items are used within domains and how this relations can be similar or different across regions.

2.2 Social Factors

The study of language contact has been studied through looking at the processes of borrowing, code-switching, and interference. Current research looks into how the intermingling of languages in a multilingual society has led to more complex language contact. With more speakers becoming multilingual, choosing a specific language for communication can also be linked to social identity (Tajfel, 1979). Therefore, in this study, a person's choice between using English, Hindi or Transliterated Hindi is not only limited to languages they know but can also extend to this theory of which social group they prefer to belong to. Factors which could contribute to this social identity could be social factors such as education, urbanization and gender. This study explicitly considers several social factors and how it impacts language use.

Urbanization: Urban areas are usually more linguistically diverse and have a higher amount of language contact. This is due to higher migration into the cities which leads to more contact between different communities. Furthermore, the more urban the city, the higher the access of to multilingual education and connectivity to the internet and multilingual media. The percentage of urban population is therefore a relevant factor in understanding the prevalence of language mixing on social media. Language contact has actually also been used to study urban cities (Chríost and Thomas, 2008; Peukert, 2013).

Literacy and Education: Higher literacy rates usually suggest an increased access to and engagement with online platforms. Furthermore, education level, especially in India since English is not everyone's native language, influences proficiency in English, impacting it's degree of use in online communication (Bhatt, 2008).

Regional Language Influence: India's diverse linguistic landscape could influence semantic variation in online communication. Khubchandani (1983) emphasizes the role of interference and code switching in multilingual communication. Therefore, we include number of Hindi speakers, whether Hindi is the 1st/2nd or 3rd language of any speakers, whether English is the 1st/2nd or 3rd language of any speakers and whether Hindi is the state's official language or not to our analysis. We do focus on these metrics as we are looking into English and Hindi data and our census data (2011) has limitations.

Gender and Language: Gender has a role to play in how language is used and changes over time are also driven by gender (Gordon, 2003; Eckert and McConnell-Ginet, 2003, 2013). Therefore, this study considers the gender distribution among Hindi speakers, sex ratio of the city and literacy rate by gender as important factors to understand the language use on social media.

3 Data

This study uses a large scale social media corpus containing 49,801,176 English tweets and 5,545,724 Hindi tweets, all originating from India.

It is important to note that the English corpus contains a huge amount of Transliterated Hindi data which is often used by people in the region especially for online communication. To analyze the data, we combined the two corpora into a single embedding space to capture semantic representations across languages. The resulting corpus contains approximately 55 million documents.

3.1 Cities

Each tweet in the corpus includes geo-location metadata, which we will use to study regional variation in lexical semantics. However, the full corpus contains data from 100 cities. To ensure we represent different regions of India and yet maintain complexity, we selected 13 cities with the highest tweet volume. Our thirteen cities are spread across India as shown in Figure 1. It is important to note that we also made sure that we chose cities with a similar level of connectivity with the internet in order to ensure uniformity. Table 1 shows the document count and regional classification for each of the selected cities.

Figure 1: Map of India with states marked for each of the thirteen chosen cities.

3.2 Semantic Domains

We analyze six semantic domains: animals, kinship, weather, professions, emotions and temporal units. These domains were selected based on their cross linguistic variation and sufficient lexical coverage in the corpus due to high usage in day-to-day life. All domains include Hindi, English and Transliterated Hindi lexical terms. The appendix mentions the whole list of lexical items used for this study.

City	Location	Count
Bengaluru	South West	2613502
Mumbai	Western Peninsular	2269945
Delhi	North	2191038
Chennai	South East	1367075
Hyderabad	South	1315379
Kolkata	East	1297674
Thane	Western Peninsular	1126080
Pune	Western Peninsular	1020440
Srinagar	North	989439
Chandigarh	North	967845
Jaipur	North West	663712
Patna	North East	559755
Lucknow	North	545069

Table 1: Count of documents from each of the thirteen chosen cities.

Table 2 shows the number of unique lexical items used per semantic domain. We ensured relatively balanced lexicons across domains to support a more robust comparison of semantic structures.

Domain	Number of Data Points
Kinship	113
Animals	282
Time	158
Professions	191
Weather	133
Emotions	138

Table 2: Number of unique lexical items for each semantic domain.

3.2.1 Kinship

Kinship is a domain in which the Hindi system is significantly more granular than the English system. Key difference include different terms for maternal vs. paternal relatives (e.g. grandparents, aunts, uncles), and specific terms for paternal uncles based on their age relative to the father. Similarly, grandchildren, nieces and nephews are also distinguished lexically based on lineage.

3.2.2 Animals

In the animal domain, Hindi often differentiates between male and female animals lexically, more extensively than in English. For example, Hindi users refer to a male cat as 'billa' (transliterated) and a female cat as 'billi' (transliterated) whereas English users typically use a gender neutral term, "cat".

3.3 Temporal Terms

This domain includes terms used for telling time such as days of the week, months of the year and terms to refer to days such as today, yesterday and tomorrow. A few notable distinctions between Hindi and English include:

1. 'kal' is used for both tomorrow and yesterday in Hindi.

2. 'parso' (transliterated) is used for both day after tomorrow and day before yesterday in Hindi.

3. For time, Hindi has specific words for 1:30 and 2:30 which do not include numerals and specific words for quarter past, half past and quarter to.

The remaining domains offer supplementary data for regional comparison, despite showing less lexical variation.

4 Methodology

The basic approach in this paper is to infer a representation of six semantic domains from population-specific corpora representing different cities in India. Once we have inferred these semantic domains, we compare them to one another in order to quantify variation and then use regression models to understand the relationship between these variations and external factors like language contact.

4.1 Inferring Semantic Domains

Because we are interested in the usage of a bilingual speech community, we combine both the Hindi (in any orthography) and the English data together. Our baseline dataset contains data from over one hundred Indian cities; this is used to infer average or non-population-specific semantic domains. Our test datasets, on the other hand, are drawn from thirteen individual cities. The idea is to compare these city-specific domains to the average domain as a means of quantifying the amount of semantic variation within these domains.

Given these corpora (the baseline corpus and the thirteen city-specific corpora), we then learn character-based embeddings using fastText in order to represent general semantic relationships between lexical items. We do not use pre-trained LLMs for creating these embeddings spaces because there is not sufficient data to do so while representing only city-level populations. Relying on models trained on outside data would risk contaminating the city-level semantic domains with information derived from the broader population.

At this stage we have distinct embedding spaces for each city-level population and for the country as a whole. The next task is to create maps or networks representing each of the six semantic domains using this embedding space. First, we manually curate the lexical items for each domain, drawing from both English and Hindi. These terms are found in Appendix A. For example, the kinship domain includes both English terms like *grandmother* and *aunt* as well as Hindi terms like *parivaar* and *mausa*. We create a network out of this domain-specific vocabulary using Jaccard similarity: each lexical item is a node in the network and the Jaccard similarity quantifies the edge weights between nodes. For instance, we would expect that *grandfather* is closer to *grandmother* than it is to *niece*. Jaccard similarity in this context is calculated by using cosine similarity to retrieve the n nearest neighbors for each word (here, $n = 1000$). High set similarity is then reflecting the fact that two words are located in the same neighborhood within the domain.

From a Saussurian perspective (de Saussure, [1916] 1983), the meaning and value of each word can be taken from the relationships within this graph. In other words, the meaning of *grandfather* is derived purely from its relationships with other items in the same kinship domain. These domains are then jointly defined by (i) using prior knowledge to select the relevant lexical items and (ii) using an embedding space to estimate edge weights.

To summarize, then, we operationalize semantic domains as networks by, first, learning a character-based embedding from each city-specific corpus and, second, using nearest neighbors in this embedding space to calculate the distance between nodes, where a node is a domain-specific lexical item. One challenge with character-based embeddings is that they can exhibit instability, reaching

different neighborhoods across multiple random initializations. We thus conduct a stability analysis to ensure that these inferred networks are reliable representations of each domain.

To test robustness, we re-ran the full pipeline for each city and each semantic domain ten times taking different sub samples of the data. Across all runs, the models consistently produced the highly similar network maps, with only minimal variation. This indicates that the inferred networks are stable.

4.2 Comparing Cities

Once we attained the Jaccard similarity between lexicon items for each domain for each city, we compared the similarity matrix between cities by calculating the mean square difference. This gave us a quantifiable difference between the structures, making it easier to group cities as being similar or different from each other. This resulted in a matrix which contained the mean square difference between the cities. We took the correlation of this matrix and then compared this between domains to see whether cities which have similar domain structures for one domain have similarity in lexicons across domains or not.

4.3 Social Features

After getting a matrix of similarities between cities across domains, we look at social features which could cause certain cities to have similar structures. Social features included the percentage of English (as a 1st, 2nd or 3rd language) and Hindi (as a 1st, 2nd or 3rd language as well as just as the mother tongue) speakers in the city, literacy rates and percentage of urban area/population. For the number of Hindi speakers and literary rates, we further got gendered data. This data is extracted from the Census of India (2011) which was published in 2018 (where city data is not available state data was used). Linear regression was conducted to see how these factors correlated to similarities/differences between the cities and the national average semantic structure.

5 Analysis

Figure 2 shows us the average correlation matrix for the mean square differences in the mappings between cities for our six domains. Here i is the taken as the national average. A correlation close to 1 shows high positive correlation shown by dark red. This signifies that the two cities had similar mean square differences for that domain suggesting a similar structure. A correlation close to -1 shows high negative correlation between the cities shown by dark blue. This shows that the cities have very different structures as their mean square differences compared to other cities in our matrix are not very similar and are quite contrasting. A value close to 0 suggests that there is no correlation between the cities. We want to observe whether there are any significant differences in the structure of semantic domains as operationalized. This would be seen if our correlation matrix has a range of values from -1 to 1 as this would suggest that each city has some difference in structure. However, if we see the same very extreme values that would suggest that all cities are correlated meaning that all structures look the same (uniformity in semantic structures). On the other hand, if we see no correlation (values just ranging near 0), that would suggest that there is no commonality in any of the structures and all cities have a very different way to portray the lexicon in the embedding space. We decided to average out and create one matrix for our analysis. This is because the matrices for each of the domains had similar values. These matrices can be seen in appendix.

5.1 Across Domains

Across domains, we see the following clusters:

1. Bengaluru, Hyderabad, Kolkata, Pune and Thane
2. Delhi, Jaipur, Lucknow and Patna
3. Chennai and Srinagar
4. Chandigarh and Mumbai
5. Mumbai and Thane

Figure 3 shows these clusters. It is important to note that the map marks the states instead of the cities to show neighboring states in an easier manner. We also see a pattern of Chennai having the most correlated structures to the national average and Pune having the least correlated structures. Our analysis shows that the domain structure changes across regions. This could be due to language contact with other languages which occurs in those states and also bleeds to neighboring states. Overall, this suggests that there is a meaningful difference in the structure of different cities and this difference is seen uniformly across domains as our clusters rarely

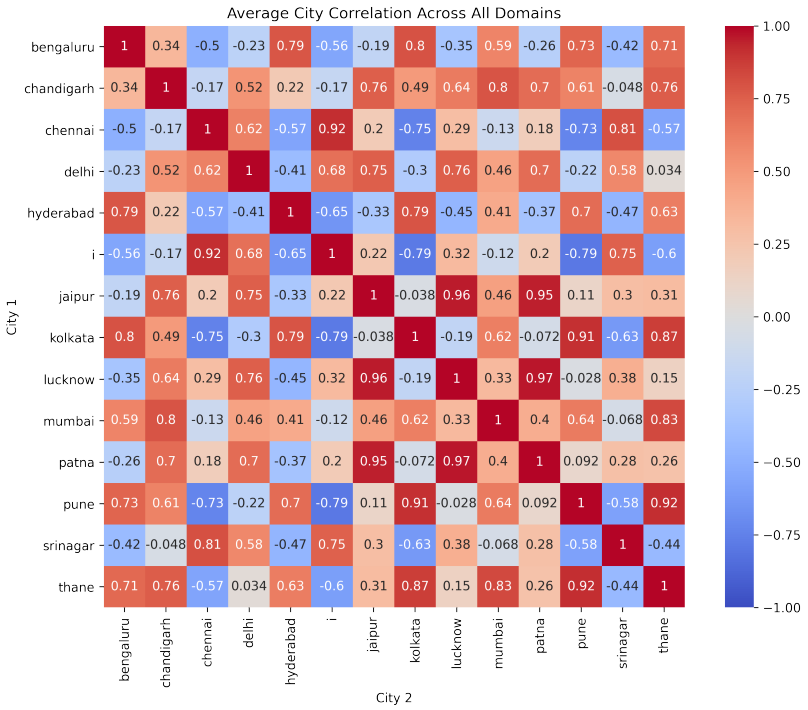


Figure 2: Average Correlation Matrix showing similarities and differences between cities across all domains. Here a value close to -1 suggests negative correlation between those cities (very different mappings) and a value close to +1 suggests positive correlation between those cities (very similar mappings). Here i refers to the national average.

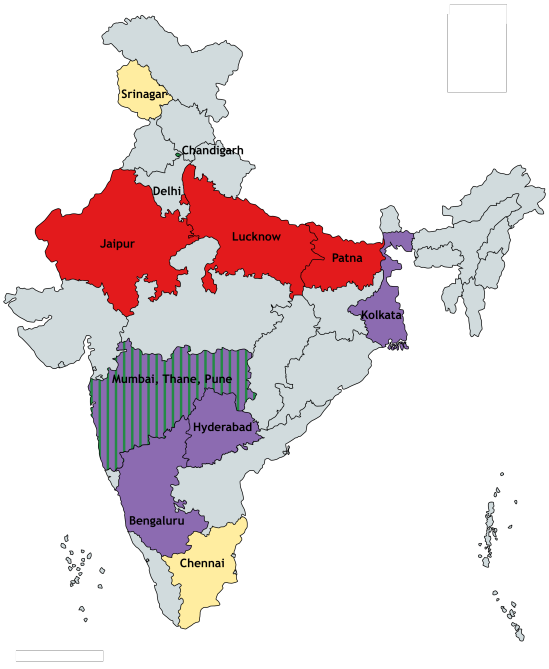


Figure 3: Groupings of positive correlation across domains on the Indian map. Here different color suggests that they have different semantic mappings and same color suggests that they have very similar semantic mappings

change in our analysis of the six domains. This suggests that geographic distance between cities impacts semantic representations.

5.2 Social Factors

We performed linear regression to examine whether a city's deviation from the national average in semantic structure could be explained by social factors. Prior to regression, all social variables were normalized to ensure comparability across scales, especially between large values (e.g. population) and percentage-based features (e.g. literacy rate). Across our domains several social features consistently contributed to predicting semantic similarity/conformity to national average in our linear regression model. These features include:

- **Literacy Rate (Overall)**: Consistently the strongest *positive* predictor across all domains. Higher overall literacy in the city is strongly associated with greater semantic conformity to the national average.
- **Literacy Rate (Male and Female)**: When overall literacy is excluded, male and female literacy show large but opposing effects - male literacy is strongly positive while female literacy is strongly negative. This suggests male

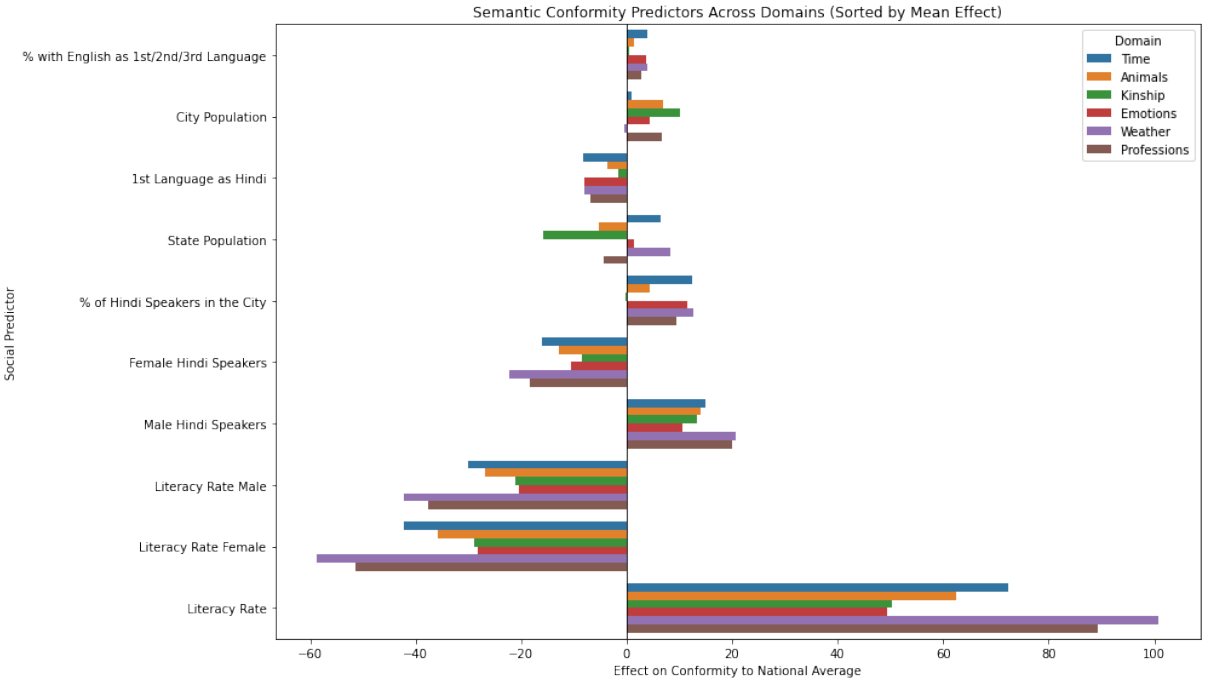


Figure 4: Social predictors of semantic conformity to the national average across domains. Positive values indicate that an increase in the feature contributes to semantic similarity with the national average.

literacy reinforces conformity, while higher female literacy correlates with divergence, consistent with the idea that women may innovate away from norms. The opposing effects also underscore multicollinearity with overall literacy.

- **Male Hindi Speakers**: Strong and consistent *positive* predictor - cities with more male Hindi speakers show greater semantic similarity to national patterns.
- **Female Hindi Speakers**: Strong *negative* predictor - possibly indicating gendered variation in language use and exposure that diverges from national norms.
- **Percentage of Hindi Speakers**: A clear *positive* influence - more Hindi presence overall contributes to semantic conformity.
- **State and City Population**: Population effects are domain -specific. Larger cities generally show a positive effect, suggesting that urban centers mirror national semantic patterns. By contrast, state population often has a negative effect in domains such as Kinship and Professions, likely reflecting the greater rural urban diversity within populous states. Thus, while cities may exert a homogenizing influence, states capture broader variation that diverges from national norms.
- **1st Language as Hindi**: Moderate *negative* effect - cities with Hindi as a first-language are somewhat less similar to the national average.
- **English (1st/2nd/3rd Language)**: Mild *positive* correlation - increased multilingualism including English is weakly associated with conformity.

These results suggest that social variables - particularly literacy, language exposure, and city/state population size - significantly shape how closely a city's semantic patterns align with the national average. The gendered contrast between male and female Hindi speakers and literacy rates, in particular, reveals complex sociolinguistic dynamics and do agree with the idea that woman diverge from norms and hence could be the drivers for language change and regional language usage.

6 Conclusion

This study investigated how the mapping between concepts and lexical items within specific semantic domains varies across geographical regions within India. Our analysis revealed that semantic mappings do vary across regions, offering insight into

language contact and multilingual variation in online communication. It showed consistent clusters of cities which had similar semantic structures. This suggests that geographic proximity influences variation in these representations. These clusters show that semantic similarity is influenced by spatial and social features. Notably, Chennai, despite not being Hindi-dominant, showed the highest similarity to the national average, while Pune showed the least, indicating that the national semantic norm reflects more than just northern, Hindi-speaking patterns.

The linear regression model showed that social features correlate closely with semantic conformity of a domain structure to the national average.

1. **Literacy rates and Number of Hindi speakers** both showed gendered divergence. High **male rates** correlated with a strong *positive* effect on semantic conformity, while **female rates** shows a strong *negative* effect. These opposing effects suggest distinct linguistic networks across gendered populations. Furthermore, it suggests that male population conforms to national average whereas female population might be driving diverse language change.

2. **State and city population effects diverge across domains**: **City population** generally shows a mild *positive* effect on semantic conformity in most domains (e.g., Kinship, Time), suggesting that larger urban centers may trend toward standardized usage. In contrast, **state population** shows a more *inconsistent* or even *negative* effect in domains like Kinship and Professions, indicating that broader regional demographics do not always align with national patterns and reflect greater internal linguistic diversity.

Overall, this study contributes to our understanding of how language contact and social features shape semantic domain structure and lexical semantics in multilingual online spaces. Our methodology - creating semantic networks from embedding spaces and enriching them with social predictors - offers a novel framework for studying semantic variation especially in multilingual settings. By offering a structured approach to examining how semantic variation aligns with regional and social characteristics in multilingual settings, this study can inform more personalized language technologies and educational resources.

References

Maria Antoniak and David Mimno. 2018. Evaluating the stability of embedding-based word similarities. *Transactions of the Association for Computational Linguistics*, 6:107–119.

Rakesh M. Bhatt. 2008. In other words: Language mixing, identity representations, and third space. *Journal of Sociolinguistics*, 12(2):177–200.

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.

Laura Burdick, Jonathan K. Kummerfeld, and Rada Mihalcea. 2021. Analyzing the surprising variability in word embedding stability across languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5891–5901, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Diarmait Mac Giolla Chríost and Huw Thomas. 2008. Linguistic diversity and the city: Some reflections, and a research agenda. *International Planning Studies*, 13(1):1–11.

Jonathan Dunn. 2023. Variation and instability in dialect-based embedding spaces. In *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pages 67–77, Dubrovnik, Croatia. Association for Computational Linguistics.

P. Eckert and S. McConnell-Ginet. 2003. *Language and Gender*. Cambridge Textbooks in Linguistics. Cambridge University Press.

Penelope Eckert and Sally McConnell-Ginet. 2013. *Language and Gender*, 2 edition. Cambridge University Press.

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.

Matthew J. Gordon. 2003. Principles of linguistic change: Social factors, volume 2. *American Anthropologist*, 105:436–437.

Gabriel Grand, Idan Asher Blank, Francisco Pereira, and Evelina Fedorenko. 2018. Semantic projection: recovering human knowledge of multiple, distinct object features from word embeddings.

Government of India. 2011. Census of india. Technical report, Government of India.

Derry Jatnika, Moch Arif Bijaksana, and Arie Ardiyanti Suryani. 2019. Word2vec model analysis for semantic similarities in english words. *Procedia Computer*

Science, 157:160–167. The 4th International Conference on Computer Science and Computational Intelligence (ICCSCI 2019) : Enabling Collaboration to Escalate Impact of Research Results for Society.

Lifeng Jin and William Schuler. 2015. A comparison of word similarity performance using explanatory and non-explanatory texts. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 990–994, Denver, Colorado. Association for Computational Linguistics.

Lachman M. Khubchandani. 1983. *Plural Languages, Plural Cultures*. University of Hawaii Press, Honolulu.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *AAAI Conference on Artificial Intelligence*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality.

Hagen Peukert. 2013. Measuring language diversity in urban ecosystems. *Linguistic Superdiversity in Urban Areas*, pages 75–95.

Abhishek Rana, Akshita Pant, Nikita Rawat, Priyanshu Rawat, Satvik Vats, and Vikrant Sharma. 2024. Semantic similarity analysis using fasttext. In *2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC)*, pages 454–460.

Ferdinand de Saussure. [1916] 1983. *Course in General Linguistics*. Duckworth, London. (trans. Roy Harris).

Henri Tajfel. 1979. An integrative theory of intergroup conflict. *The social psychology of intergroup relations/Brooks/Cole*.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2020. Sentiment lexicon expansion using word2vec and fasttext for sentiment prediction in tamil texts. *2020 Moratuwa Engineering Research Conference (MERCon)*, pages 272–276.

A Appendix

A.1 Lexicon

A.1.1 Kinship Terms

English: grandmother, grandfather, aunt, uncle, mother, father, sister, brother, niece, nephew, daughter, son, granddaughter, grandson, cousin, husband, wife, father-in-law, mother-in-law, brother-in-law, sister-in-law, children, brother-in-law's wife, son-in-law, daughter-in-law

Transliterated Hindi: parivaar, nani, nana, dadi, masi, mausa, mummy, pita, papa, bua, chacha, bhua, tau, tauji, behen, bhai, didi, bhaiya, bhaanji, bhaanja, bhajiti, bhatija, beti, beta, naatin, naati, pota, poti, pati, patni, sasur, saas, devar, nanad, bacche, jeeja, devrani, saala, daamaad, bahu

Hindi: नानी, नाना, दादी, दादा, मासी, मौसी, मामा, मम्मी, माँ, पिता, पापा, बुआ, चाचा, ताऊ, बड़े पापा, बहन, भाई, दीदी, भैया, भांजी, भांजा, भतीजी, भतीजा, बेटी, पुत्री, बेटा, पुत्र, नतिनी, नातिन, नाती, पोती, पोता, पति, पत्नी, ससुर, सास, देवर, ननद, बच्चे, जीजा, देवरानी, साला, दामाद, बहू

A.1.2 Animal Terms

English: Hyena, Dove, goat, Snail, monkey, Mosquito, Crocodile, Lizard, Earthworm, camel, Tortoise, Myna, Turtle, Fish, Caterpillar, Bugs, Birds, Deer, Leopard, Lioness, Sheep, Goose, Pig, Wolf, Seahorse, Bat, mouse, Insect, Bear, Panther, Sealion, Fox, Donkey, Spider, Housefly, elephant, Honeybee, Butterfly, Snake, Gander, Cuckoo, Mongoose, Buffalo, Grasshopper, Hen, Lion, Animal, Aquatic, Kite, Weaverbird, Rabbit, Duck, Alligator, Woodpecker, Chameleon, Squirrel, Eagle, Octopus, Cricket, Pet, Guinea pig, Cow, Giraffe, Tiger, Tigress, Pigeon, Prawns, Whale, Dolphin, dog, Horse, Bird, Shark, Hawk, Parrot, Insects, Hippopotamus, Owl, cat, Jellyfish, Oyster, Mammals, Vulture, Cockroach, Ant, Frog, Crow, Rooster, Wild

Transliterated Hindi: tidda, lomdi, chipakali, madhumakkhii, jiraaf, shernii, hiran, sher, ghonghaa, totaa, safed kabuuTar, kiida, bhaalu, mendhak, gae, chidiyaa, suar, bakri, hathinii, saanp, chuuhaa, haangar, ashtabahu, kauvaa, battakh, murgii, chuhiya, lakadbagghaa, baaz, jhiingur, gini pig, titli, bhed, billi, kabuuTar, paaltoo, bandariya, bakraa, jeliifish, ullu, jhiinga machhli, gauraiyaa, tilacchattaa, vhel, jalsinh, samudri ghodaa, oont, makri, girgit, kharagosh, oontnii, bhediyaa, siip, giddh, daryaai ghodaa, haathi

Hindi: घोंघा, भेड़, पालतू, कीड़ा, हाथी, कछुआ, जंगली, साँप, मुर्गा, ऊँटनी, लोमड़ी, मक्खी, सियार, केंचुआ, तिलचट्टा, घड़ियाल, गधा, चुहिया, बंदरिया, हंसिनी, मधुमक्खी, ख़रगोश, घोड़ा, बया, बकरी, गिद्ध, बिल्ली, कीड़े, बाघ, भालू, जेलीफ़िश, बंदर, तितली, ऊँट, मकड़ी, गिरगिट, बकरा, झींगा मछली, व्हेल, कुत्ता, भेड़िया, हाँगर, चील, चूहा, मैना, तेंदुआ, बत्तख़, गौरैया, समुद्री घोड़ा, भैंस, शेरनी, बिल्ला, गाय, मेंढक, मछली, गिनी पिग, चींटी, कबूतर, जिराफ़, मच्छर, लकड़बग्घा, तारामीन, गिलहरी, छिपकली, जानवर, कुतिया, मुर्गी, सुअर, मगरमच्छ, हथिनी, शेर, दरियाई घोड़ा, जलसिंह, चिड़िया, हिरण, इल्ली, कठफोड़वा, कौवा, अष्टबाहु, झींगुर, नेवला, कोयल, तोता, हंस, स्तनधारी, समुद्री, सूंस, बाघिन, सफ़ेद कबूतर, चमगादड़, सीप, बाज, उल्लू, टि–

ड्डा

A.1.3 Time Terms

English: Second, Minute, Hour, Day, Week, Month, Year, Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, January, February, March, April, May, June, July, August, September, October, November, December, Morning, Afternoon, Evening, Night, Midnight, Yesterday, Today, Tomorrow, Day before yesterday, Day after tomorrow, Now, Later, Earlier, O'clock, Half past, Quarter past, Quarter to, Always, Often, Sometimes, Rarely, Never, For a short time, For a long time, Since, Until

Transliterated Hindi: sekand, minat, ghanta, din, saptah, saptaah, mahina, saal, varsh, ravivar, somvar, mangalvar, budhvar, guruvaar, shukravar, shanivar, janavari, pharavari, march, aprail, joon, julai, agast, sitambar, aktoobar, navambar, disambar, subah, dophar, shaam, raat, aadhi raat, kal, aaj, kal, parson, abhi, baad mein, pehle, baje, saade, sava, paune, hamesha, aksar, kabhi-kabhi, shayad hi kabhi, kabhi nahi, thodi der ke liye, lambe samay tak, se, tak

Hindi: सेकंड, मिनट, घंटा, दिन, सप्ताह, महीना, साल, रविवार, सोमवार, मंगलवार, बुधवार, गुरुवार, शुक्रवार, शनिवार, जनवरी, फरवरी, मार्च, अप्रैल, मई, जून, जुलाई, अगस्त, सितंबर, अक्टूबर, नवंबर, दिसंबर, सुबह, दोपहर, शाम, रात, आधी रात, कल, आज, कल, परसों, परसों, अभी, बाद में, पहले, बजे, साढ़े, सवा, पौने, हमेशा, अक्सर, कभी–कभी, शायद ही कभी, कभी नहीं, थोड़ी देर के लिए, लंबे समय तक, से, तक, वर्ष

A.1.4 Weather Terms

English: Sun, Rain, Wind, Snow, Cloud, Weather, Hot weather, Cool weather, Pleasant weather, Weather change, Weather forecast, Seasons, Spring, Winter, Summer, Autumn, Rainy, Temperature, Hot, Humid, Cold, Moisture, Scorching, Sunshine, Sunrise, Sunset, Sky, Cloudy, Rainbow, Drizzle, Storm, Cyclone, Lightning, Fog, Dew, Snowfall, Hail

Transliterated Hindi: Sooraj, Baarish, Hawaa, Baraf, Baadal, Mausam, Garam Mausam, Thanda Mausam, Suhaana Mausam, Mausam Parivartan, Mausam Purvaanumaan, Rituyen, Vasant Ritu, Sardi, Thand, Shishir, Sheet Ritu, Grishm Ritu, Patjhad, Sharad Ritu, Barsaat, Varsha Ritu, Hemant Ritu, Taapmaan, Paara, Aardrataa, Thandak, Nami, Chilchilaatii, Dhuup, Suryodaya, Suryaast, Aasmaan, Aakaash, Boondaabaandi, Phuhaar, Tufaan, Chakravaat, Bijli, Kohraa, Os, Barfbari, Ola Vrishti

Hindi: सूरज, बारिश, हवा, बरफ, बादल, मौसम, गर्म मौसम, ठंडा मौसम, सुहाना मौसम, मौसम परिवर्तन, मौसम पूर्वानुमान, ऋतुएं, मौसम, वसंत ऋतु, सर्दी, ठंड, शिशिर, शीत ऋतु, गर्मी, ग्रीष्म ऋतु, पतझड़, शरद् ऋतु, बरसात, वर्षा ऋतु, हेमंत ऋतु, तापमान, पारा, गर्मी, उमस, आर्द्रता, ठंडक, नमी, चिलचिलाती, धूप, सूर्योदय, सूर्यास्त, आसमान, आकाश, बदली, इंद्रधनुष, बूंदाबांदी, फुहार, तूफ़ान, चक्रवात, बिजली, कोहरा, ओस, बर्फ़बारी, ओला वृष्टि

A.1.5 Emotion Terms

English: blissful, brave, careful, cautious, clever, curious, excited, friendly, glad, good, great, happy, innocent, interesting, optimistic, pleasant, pleased, proud, quiet, satisfied, sensible, serious, surprised, angry, arrogant, awful, bad, bored, crazy, disappointed, exhausted, frightened, sad, guilty, helpless, hurt, lonely, mad, miserable, nervous, shocked, sheepish, silly, strange, terrible, upset

Transliterated Hindi: anandmay, bahaadur, saavadhan, satark, chaalak, utsuk, uttejit, mitravat, prashann, achcha, mahaan, khush, nirdosh, dilchasp, aashavaadi, sukhad, santusht, garvit, shaant, samajhdaar, gambhir, haeraan, naaraaj, abhimaani, daraavana, bura, uba hua, sanki, niraash, thaka, bhayabhit, dukhi, doshi, asahaay, aahat, akela, paagal, abhaaga, ghabaraaya hua, haeran, sharminda, murkh, anokha, bhayaanak, pareshan

Hindi: आनंदमय, बहादुर, सावधान, सतर्क, चालाक, उत्सुक, उत्तेजित, मित्रवत, प्रसन्न, अच्छा, महान, खुश, निर्दोष, दिलचस्प, आशावादी, सुखद, सन्तुष्ट, गर्वित, शांत, संतुष्ट, समझदार, गंभीर, हैरान, नाराज, अभिमानी, डरावना, बुरा, ऊबा हुआ, सनकी, निराश, थका, भयभीत, दुखी, दोषी, असहाय, आहत, अकेला, पागल, अभागा, घबराया हुआ, हैरान, शर्मिंदा, मूर्ख, अनोखा, भयानक, परेशान

A.1.6 Profession Terms

English: Butcher, Florist, Travel agent, Scientist, Gardener, Mason, Pilot, Librarian, Model, Shop assistant, Bus driver, Real estate agent, Lawyer, Cook, Fireman, Poet, Poetess, Soldier, Receptionist, Designer, Fire fighter, Fisherman, Waitress, Actress, Author, Dentist, Shop keeper, Traffic warden, Baker, Journalist, Judge, Actor, Plumber, Secretary, Veterinary doctor, Farmer, News reader, Craftsman, Lifeguard, Photographer, Taxi driver, Carpenter, Optician, Accountant, Teacher, Electrician, Postman, Tailor, Painter, Policeman, Engineer, Hairdresser, Policewoman, Nurse, Doctor, Mechanic, Translator, Politician, Lecturer, Waiter, Workers,

Cleaner, Pharmacist

Transliterated Hindi: Machuaara, Anuvaadak, Chashma Banane Wala, Phoolwala, Naanbai, Sachiv, Shramik, Samachar Paadak, Vaastukar, Sramik, Model, Nalsaaj, Maarjak, Sipaahi, Svaagati, Lekhak, Kavi, Vakil, Aushadhajny, Badai, Baayara, Abhinetri, Yaatra Agent, Abhiyanta, Bas Chalak, Daakiya, Vigyaanik, Sainik, Dukan Sahayak, Sharir Raksak, Rajnitigy, Rajgir, Viman Chalak, Granthaagarik, Bhumi Bhavan Abhikarta, Nyaayadhish, Chitrkar, Abhineta, Kasaai, Mechanic, Shikshak, Dukandar, Shilpkar, Naai, Yaatayaat Nirikshak

Hindi: बायरी, कवियित्री, डाकिया, पत्रकार, वकील, मॉडल, अग्निशामक कर्मचारी, दंत चिकित्सक, दुकानदार, नानबाई, भूमि भवन् अभिकर्ता, रसोइया, बिजली मिस्त्री, बढ़ई, मार्जक, राजगीर, नलसाज, कवि, ग्रंथागारिक, रूपकार, अभिनेत्री, वैज्ञानिक, मुनीम, औषधज्ञ, विमान चालक, बायरा, मैकेनिक, सचिव, दुकान सहायक, स्वागती, यात्रा एजेंट, नाई, अभिनेता, चित्रकार, मछुआरा, माली, शिल्पकार, फूलवाला, लेखक, समाचार पाठक, नर्स, यातायात निरीक्षक, फोटोग्राफर, शरीर रक्षक, अनुवादक, पशु चिकित्सक, श्रमिक, किसान, वास्तुकार, महिला सिपाही, दर्जी, टैक्सी चालक, शिक्षक, अभियन्ता, कसाई, राजनीतिज्ञ, सैनिक, चिकित्सक, बस चालक, चश्मा बनाने वाला, न्यायाधीश, सिपाही, व्याख्याता

A.1.7 Domain wise Analysis

A.2 Domains

Fig. 5 shows the correlation matrices across the thirteen cities for all six domains.

A.2.1 Time

Time matrix shows positive correlation between cities of (taking a cut off of 0.81)

1. Bengaluru, Hyderabad, Kolkata, Pune and Thane
2. Chandigarh and Mumbai
3. Chennai and Srinagar
4. Delhi, Patna, Lucknow and Jaipur

Mumbai and Thane also have high positive correlation but the other members of those groups do not. Compared to our average (India), Chennai and Delhi have high positive correlation and Hyderabad and Kolkata have high negative correlation.

A.2.2 Weather

Weather matrix shows positive correlation between cities of (taking a cut off of 0.81)

1. Bengaluru, Hyderabad, Kolkata, Pune and Thane
2. Chandigarh and Mumbai
3. Delhi, Patna, Lucknow and Jaipur

Again, Mumbai and Thane also have high positive correlation but the other members of those groups do not. Chennai has high negative correlation with (a) and Srinagar has no strong correlations with any groups. Compared to our average (India), Chennai has high positive correlation and Hyderabad, Bangalore, Pune and Kolkata have high negative correlation.

A.2.3 Animals

Animals matrix shows positive correlation between cities of (taking a cut off of 0.81)

1. Bengaluru and Kolkata
2. Chandigarh, Pune and Thane
3. Patna, Lucknow and Jaipur

Again, Mumbai and Thane also have high positive correlation but the other members of those groups do not. Compared to our average (India), Chennai has high positive correlation and Pune has high negative correlation.

A.2.4 Kinship

Kinship matrix shows positive correlation between cities of (taking a cut off of 0.81)

1. Kolkata, Pune and Thane
2. Patna, Lucknow and Jaipur

Again, Mumbai and Thane also have high positive correlation but the other members of those groups do not. Compared to our average (India), Chennai has the highest positive correlation and Pune has highest negative correlation.

A.2.5 Emotions

Emotions matrix shows positive correlation between cities of (taking a cut off of 0.81)

1. Bengaluru, Hyderabad, Kolkata, Pune and Thane

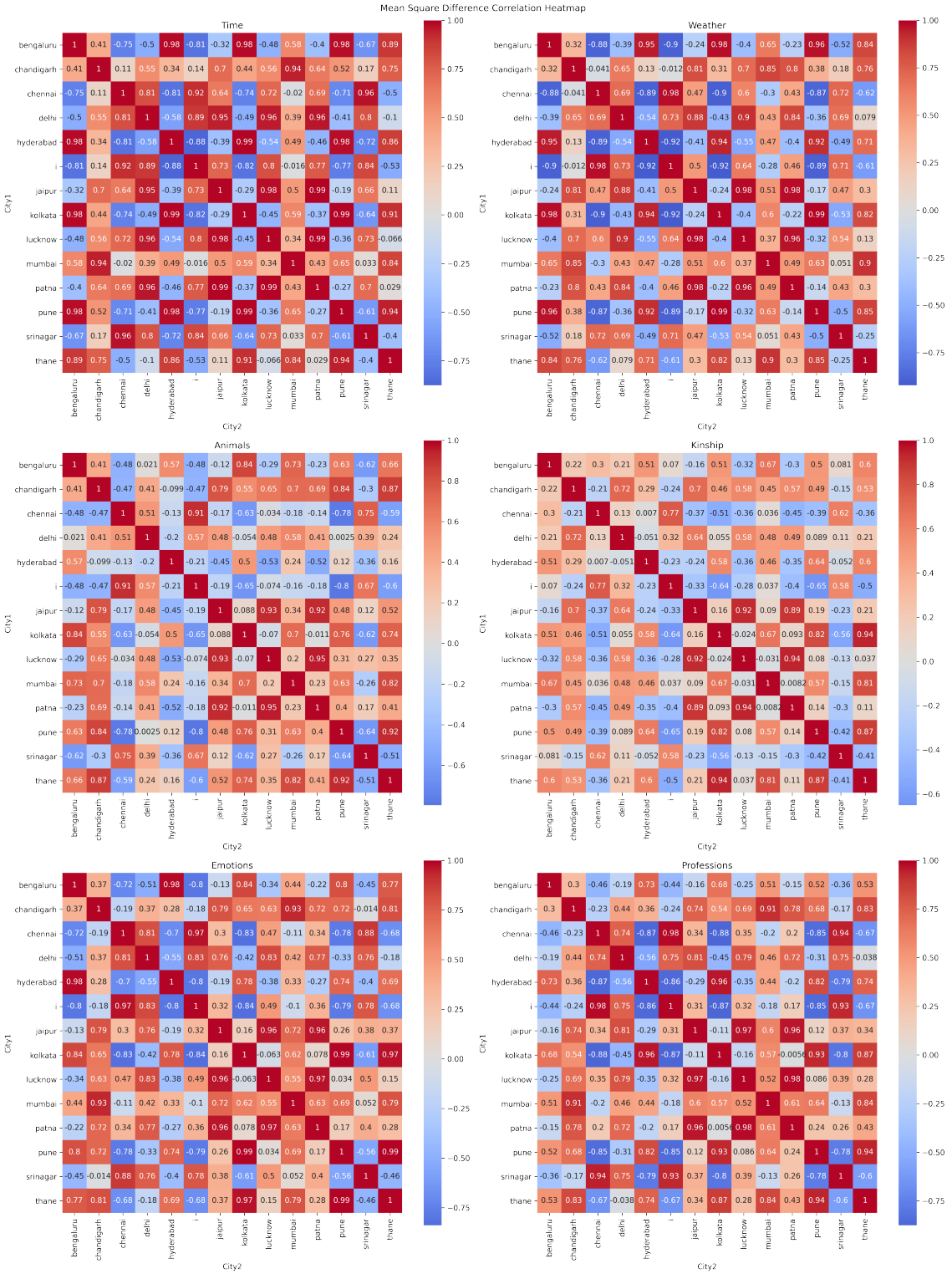


Figure 5: Correlation matrices of mean square differences for all domains. Here a value close to -1 suggests negative correlation between those cities (very different mappings) and a value close to +1 suggests positive correlation between those cities (very similar mappings)

2. Chandigarh and Mumbai, Thane

3. Chennai and Srinagar, Delhi

4. Delhi, Jaipur, Lucknow and Patna

Compared to our average (India), Chennai and Delhi have high positive correlation and group (a) have high negative correlation.

A.2.6 Professions

Professions matrix shows positive correlation between cities of (taking a cut off of 0.81 with atleast two cities of the group)

1. Hyderabad, Kolkata, Pune and **Thane**

2. Chandigarh, Mumbai and **Thane**

3. Chennai and Srinagar

4. Delhi, Jaipur, Lucknow and Patna

Bengaluru does not have any high correlation with other cities. Compared to our average (India), Chennai and Srinagar have high positive correlation and Pune, Kolkata and Hyderabad have high negative correlation.

Disentangling lexical and grammatical information in word embeddings

Li Liu
OLST, Université de Montréal
Montréal, Canada
li.liu.2@umontreal.ca

François Lareau
OLST, Université de Montréal
Montréal, Canada
francois.lareau@umontreal.ca

Abstract

To enable finer-grained linguistic analysis, we propose a method for the separation of lexical and grammatical information within contextualized word embeddings. Using CamemBERT embeddings for French, we apply our method to 14,472 inflected word forms extracted from the French Lexical Network (LN-fr), covering 1,468 nouns, 202 adjectives, and 299 verbs inflected via 14 distinct grammatical feature values. Our iterative distillation process alternates two steps until convergence: (i) estimating lexical or grammatical vectors by averaging the embeddings of words that share the same lexeme or grammatical feature value, and (ii) isolating the complementary component of each word embedding by subtracting the estimated vector. To assess the quality of the decomposition, we measure whether the resulting lexical and grammatical vectors form more compact clusters within their respective groups and whether their sum better reconstructs the original word embeddings. All evaluations rely on euclidean (L2) distance. The observed improvements in both clustering and reconstruction accuracy demonstrate the effectiveness of our approach.

1 Introduction

Static word embeddings, such as those generated by *word2vec* (Mikolov et al., 2013b,a), assign a single, fixed vector to each word form based on its general contextual usage. This approach conflates distinct meanings of polysemous words or homonyms and fails to capture morphological compositionality, as it does not model how word forms may share a common core lexical meaning or how affixes encode grammatical features. For morphologically rich languages, this entanglement can hinder fine-grained linguistic analysis. Previous work, such as Lareau et al. (2015), addressed this issue by proposing a method to decompose static embeddings into lexical and inflectional components, aiming to obtain semantically purer representations.

Contextualized embeddings from pretrained language models such as BERT (Devlin et al., 2019) produce dynamic, context-sensitive vectors that implicitly encode a range of linguistic information, including morphology and syntax. This has substantially improved the modeling of polysemy, homonymy, and morphosyntactic variation compared to static embeddings. However, it remains unclear how lexical and grammatical features are represented within these embeddings and whether they can be meaningfully disentangled. In this paper, we revisit the problem of separating lexical and grammatical information in word embeddings, focusing on embeddings produced by CamemBERT (Martin et al., 2020) for French, a language with a relatively rich morphology.

This work was originally motivated by a separate study where we aimed to measure the semantic idiomaticity of French idioms. Semantic idiomaticity refers to the extent to which the meaning of an idiom cannot be inferred from its component words. While CamemBERT is able to distinguish free simple lexemes from words within idioms, it struggles with component words within idioms of different levels of semantic idiomaticity (Liu and Lareau, 2024). This suggests that the model captures idiomaticity at a superficial lexical level, but is not sensitive to the internal semantic structure of idioms. We hypothesized that this limitation is due to the entanglement of multiple types of idiomaticity, not only semantic, but also morphological and syntactic. In order to isolate purely semantic meaning from grammatical interference, we turned to the problem of disentangling lexical and grammatical components in contextual embeddings. The current study develops and evaluates a method for this task, inspired by the methodology proposed by Lareau et al. (2015).

We assume that a word embedding can be modeled as the linear combination of two components, a lexical vector capturing its core lexical meaning,

Proceedings of the 16th International Conference on Computational Semantics, pages 331–340
September 22-23, 2025, Licensed under the Creative Commons Attribution 4.0 International License

and a grammatical vector encoding morphosyntactic information. Therefore, we should be able to isolate one component by subtracting the other. Our method relies on two assumptions:

1. All inflected forms of a lexeme share a common core lexical meaning.
2. All words inflected via the same grammatical feature value (i.e., all plural nouns, or all feminine adjectives) share a common grammatical meaning, regardless of allomorphy.

Under this framework, the lexical vector of a lexeme can be estimated either directly, by averaging the embeddings of its inflected forms, or indirectly, by subtracting a shared grammatical vector from each word embedding. Likewise, grammatical vectors associated with specific feature values can be derived by averaging over relevant word embeddings or by removing lexical content.

To obtain more accurate and disentangled representations, we develop an iterative distillation process that integrates both estimation strategies. At each step, one component is isolated by subtracting the current estimate of the other, then refined by averaging over the pertinent group of words (e.g., all inflected words of a lexeme, or all words sharing a grammatical feature value). This process incrementally improves both components over successive iterations.

We hypothesize that, after distillation, lexical vectors belonging to the same lexeme on the one hand, and grammatical vectors sharing the same feature value on the other, get closer in the vector space. We evaluate this by comparing the average pairwise L2 distances within each group before and after distillation. We also assess the reconstruction accuracy of the original embeddings by measuring the difference between each embedding and the sum of its distilled lexical and grammatical components.

In this study, we focus specifically on inflection and leave aside derivation, as it is often non-compositional. We worked on French because it has a sufficiently rich morphology for it to be non-trivial, and we had access to the data we needed. However, our method is language-agnostic, and such data is relatively easy to come by for a variety of languages.

2 Related work

Recent studies have highlighted that contextualized word embeddings encode various types of linguistic information in a high entangled form (López-Otal et al., 2025; Ravfogel et al., 2020). This has sparked growing interest in disentangling grammatical information. However, most existing work addresses this challenge in the context of downstream tasks or model performance, rather than focusing on extracting grammatically meaningful representations for linguistic analysis (Huang et al., 2021; Li et al., 2021; Chen et al., 2019; Ravfogel et al., 2020; Omrani Sabbaghi and Caliskan, 2022).

To our knowledge, few studies have explicitly addressed this question from the perspective of linguistic analysis. The work most closely related to ours that we know of is by Lareau et al. (2015), who developed a method applied to decompose static *word2vec* embeddings in English. Their approach, based on averaging and subtraction, was tested on a small-scale dataset of around 20 verbs, with a primary focus on lexical vectors. Their method struggled with homonyms due to the static nature of *word2vec* embeddings. In contrast, our approach leverages contextualized embeddings, which mitigate this issue. It is also applied to a much larger and more diverse natural corpus. While inspired by their methodology, we extend it with an iterative refinement process and expand the analysis to include grammatical vectors as well. In addition, we introduce a broader set of evaluation metrics.

3 Experiment

3.1 Data

For our experiment, we used data from French Lexical Network (LN-fr) v3 (Polguère, 2009; Lux-Pogodalla and Polguère, 2011; Polguère, 2014; ATILF, 2023), an open-access lexical database manually developed according to the methodological principles of explanatory combinatorial lexicology (Mel'čuk, 2006). Each entry in LN-fr represents a disambiguated lexical unit in French, corresponding to a distinct and well-defined sense of a simple lexeme or an idiom. In our study, we focused exclusively on simple lexemes (hereafter referred to as lexemes). Each lexeme has a part of speech (POS) tag; since we studied inflectional types in French, we extracted only the nouns, adjectives, and verbs, other classes being invariant.

Each lexeme is associated with one or more lexicographic examples sourced from corpora. These examples were carefully selected by lexicographers to reflect real-world usage, showcasing the syntax, semantics, and combinatorial properties of the

lexemes (Lux-Pogodalla, 2014). Furthermore, the annotation explicitly identifies the position of the words corresponding to the lexeme within each example. These words represent inflected forms of the lexeme in the sentence. A single lexeme may be associated with multiple words within an example. This can occur through repetitions or analytic forms (such as past tense, e.g., *ai mangé* '(I) have eaten'). To simplify the analysis, such cases were excluded. Only examples containing a single word corresponding to a lexeme were retained.

Most grammatical features are not annotated in LN-fr, with the exception of number and gender. We therefore used Stanza (Qi et al., 2020) to analyze the examples of the lexemes and complete the annotation of their remaining grammatical features. For number and gender, we compared the LN-fr annotations with those produced by Stanza and found them to be fully consistent. Given that nouns and adjectives, the main categories marked for these features, account for over 86% of our data, this consistency supports the reliability of Stanza for morphological annotation and indicates that our method should work even without annotated data. To further reduce potential errors, we compared the POS tags and lemmas returned by Stanza with the manual annotations in LN-fr, removing 230 lexemes where the POS assignments did not match.

We generated word embeddings for lexemes in our data using CamemBERT (Martin et al., 2020) for our experiment. CamemBERT is a pretrained contextualized language model for French, where each token in the sentence is represented differently depending on the other tokens in the context. We used the representations from the last layer. For words tokenized into sub-word tokens, we sum all sub-word embeddings to get the word's embedding. We used example sentences retrieved from LN-fr as context and generated vectors that represent the inflected forms of lexemes. We considered only lexemes with at least four examples in the database. This ensures more stable and representative lexical embeddings by averaging over multiple contexts and helps achieve better coverage of a lexeme's inflectional paradigm. Moreover, to reduce model-internal bias, we applied mean-centering to all embeddings, removing common components unrelated to lexical or grammatical distinctions.

In total, we extracted from LN-fr nearly 2,000 lexemes, with significantly more nouns than adjectives or verbs. Each lexeme is accompanied by its word forms, example sentences, along with corresponding word embeddings and annotated grammatical information. Table 1 summarizes the number of words associated with each feature value within each grammatical category. We group nouns, adjectives, and verbs according to the grammatical categories they express: nouns by number, adjectives by number and gender, and verbs by number, tense, mood, finiteness, voice, gender and person. Words lacking relevant annotation are excluded from the count, and only feature values with at least 200 words across lexemes are retained to ensure sufficient data for reliable analysis.

	Lexemes	**Words**
Noun	**1468**	**11159**
sing		8716
plur		2443
Adjective	**202**	**1344**
sing		1038
plur		306
masc		837
fem		507
Verb	**299**	**1969**
sing		834
plur		359
pres		903
imp		245
ind		1159
inf		715
fin		1182
per-3		1056
Total	**1969**	**14472**

Table 1: Lexemes and words (counted by grammatical feature values) in our dataset. Abbreviations for grammatical feature values: *sing*=singular, *plur*=plural, *masc*=masculine, *fem*=feminine, *pres*=present, *imp*=imperfect, *ind*=indicative, *inf*=infinitive, *fin*=finite, *per-3*=third person.

3.2 Methodology

Relying on the assumptions outlined in §1, we propose an iterative distillation method to decompose word embeddings into two components: a **lexical vector** representing its core lexical meaning and a **grammatical vector** encoding morphosyntactic information.

Let L denote a lexeme with n observed inflected forms $\{w_1, w_2, \ldots, w_n\}$. These forms share a common lexeme vector, estimated by the average of their word embeddings:

$$\mathbf{L} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{w}_i$$

Similarly, for a grammatical feature value G that appears in m words $\{w_1, w_2, \ldots, w_m\}$ in our dataset, we define a shared grammatical vector as:

$$\mathbf{G} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{w}_i$$

For example, the lexeme *eat* includes words such as *eat*, *eats*, *ate*, etc., whose embeddings share a lexical component $\overrightarrow{eat}$. The nominal number feature PLUR applies to words like *apples*, *bikes* and *houses*, whose embeddings share a grammatical component $\overrightarrow{\text{PLUR}_\text{N}}$.

Let w be a word comprising a lexical base l and a grammatical feature value g, its word embedding can be approximated as the sum of two components:

$$\mathbf{w} \approx \mathbf{l}_w + \mathbf{g}_w$$

To obtain purer lexical and grammatical embeddings, we apply two update steps:

Lexical vector update For each w of a lexeme L, we initialize its local grammatical vector g_w using its feature value vector $\mathbf{G}$. Then we subtract this grammatical vector to isolate its current lexical vector l_w. For each lexeme L, we average all its words' l_w to update the lexeme vector $\mathbf{L}$.

$$\mathbf{l}_w = \mathbf{w} - \mathbf{G}, \quad \mathbf{L} \leftarrow \frac{1}{|L|} \sum_{w \in L} \mathbf{l}_w$$

Grammatical vector update Likewise, for each w, we can initialize its local lexical vector l_w using its lexeme vector $\mathbf{L}$. Then by subtracting this l_w from the word embedding, we get the word's current grammatical vector g_w. The average of g_w of all words inflected via G is calculated to update $\mathbf{G}$.

$$\mathbf{g}_w = \mathbf{w} - \mathbf{L}, \quad \mathbf{G} \leftarrow \frac{1}{|G|} \sum_{w \in G} \mathbf{g}_w$$

Our approach is an iterative process that alternates between the two updates, where the output of one step serves as the input for the next. The process continues until the difference between successive updates becomes negligible—typically after just five or six iterations.

To distill lexical or grammatical vectors of a word, we can either start with the lexical vector update by estimating $\mathbf{G}$, or with the grammatical vector update by estimating $\mathbf{L}$. As subtraction is central to both steps, the quality of the initial estimate is critical: any noise in the subtracted vector propagates into the result. Thus, the more accurate the initial estimate, the better the decomposition. It should be stressed that this initial estimate is not random; it is the mean of a set of vectors, and thus yields the same result every time.

We find that initializing with grammatical vectors is more robust. When estimating a grammatical vector $\mathbf{G}$ for a feature value (e.g., PLUR_N), we average the embeddings of all words (e.g., *apples*, *bikes*, *houses*, etc.) that share this feature. Since these words are typically lexically diverse, their lexical components tend to cancel each other out, resulting in a relatively clean approximation of the grammatical meaning. While estimating a lexeme vector $\mathbf{L}$ (e.g., for lexeme *eat*), we average a small number of words inflected from the lexeme (e.g., *eat*, *eats*, *ate*). These words often differ in grammatical properties and appear in different contexts, introducing noise that can distort the estimate of their core lexical meaning.

Furthermore, we extend this idea to handle multi-feature grammatical information, which is common in French. While nouns typically carry only one grammatical feature NUMBER (SING or PLUR), adjectives and verbs express multiple features simultaneously. Specifically, adjectives reflect both number and gender, while verbs can encode tense, mood, person, number, etc. When extracting a word's lexical vector, we remove a composite grammatical vector that corresponds to the full set of feature values it carries. This vector is estimated by averaging the embeddings of all words sharing the exact same feature combination. Conversely, in extracting grammatical vectors, we isolate each feature value independently. For example, to estimate the vector for present tense, we use all verb forms that express the present tense, regardless of their other grammatical properties. This targeted averaging provides a clearer estimate of the intended grammatical dimension.

In summary, our method alternates between subtracting a full grammatical vector to refine lexical vectors, and subtracting the current lexical vector to isolate individual grammatical components. We apply this procedure to verbs, nouns, and adjectives grouped by their feature values, iterating until convergence.

3.3 Evaluation metrics

To assess the effectiveness of this method, we adopt two complementary evaluation metrics.

Internal distance Our evaluation is grounded in the assumptions outlined earlier: (1) words belonging to the same lexeme differ only in their grammatical components, and (2) words that share the same grammatical components differ primarily in their lexical meaning. From this, we hypothesize that once the grammatical components are removed from the original embeddings, the remaining lexical vectors should form tighter clusters within each lexeme group, because what is left should be close to the naked lexical information. Similarly, if the lexical component is subtracted, the remaining grammatical vectors should show greater internal consistency within each feature value group. To verify this, we measure the internal compactness of each group before and after distillation.

For each lexeme, we calculate the average pairwise distance among the embeddings of its inflected words prior to distillation. We then repeat the measurement using only the lexical components l_w extracted from these words after distillation. A reduction in distance suggests that the lexical content has been more effectively isolated.

Similarly, for each feature value, we first calculate the average pairwise distance among the original embeddings of all words marked with that value. We then calculate the same measure using only the grammatical vectors g_w corresponding to the feature value isolated from those words. A tighter clustering in this space would indicate that the shared morphosyntactic property has been captured more clearly.

As a baseline, we compute the average pairwise distance between random word pairs that do not share either a lexeme or any grammatical feature values, applying the same subtraction procedure. For each run, we sample up to 10,000 such random pairs; if the total number of admissible pairs is smaller, we use all available pairs. We repeat this process 10 times and report the mean across runs. Since these words are unrelated in both lexical and grammatical dimensions, their vectors should not become closer after distillation. This allows us to verify that any observed distance reduction in groups defined by shared lexemes or feature values is not merely an artifact of the subtraction process, but reflects meaningful linguistic structure.

Reconstruction accuracy We evaluate whether the lexical and grammatical components can faithfully reconstruct the original word embeddings. For each word, we compare its original embedding with two reconstructed versions: one using the initial estimates of its lexical vector and the grammatical vectors corresponding to its set of feature values, and another using the distilled vectors obtained. A lower reconstruction error in the latter case implies improved preservation of the original embeddings' structure.

3.4 Distance metric

Both the distillation process and the subsequent evaluation require a way to quantify how the vectors change under our distillation method. To compare these vectors, we initially calculated both cosine similarity and L2 distance.

Cosine similarity is commonly adopted as a metric for semantic similarity in natural language processing (NLP), as it captures the angular relationship between vectors while ignoring their magnitude. However, our method involves vector subtraction, which can substantially alter both direction and length. This makes cosine similarity potentially misleading: in extreme cases, two vectors may retain the same angle (i.e., yield a high cosine similarity) while differing greatly in magnitude, making them appear semantically close even when they are not. Previous studies have also shown that cosine similarity can be distorted in contextualized embedding models due to anisotropy and frequency effects (Ethayarajh, 2019; Timkey and van Schijndel, 2021; Zhou et al., 2022).

In our evaluation, cosine similarity and L2 distance often led to divergent conclusions. Since L2 distance captures both angular and magnitude-related differences, we consider it to be a more reliable indicator of the structural changes introduced by our method. Furthermore, we found no strong theoretical reason to prefer cosine similarity in our setting beyond its popularity in previous work. Given these considerations, we focus exclusively on L2 distance in the results reported below.

4 Results and Discussion

In this section, we evaluate whether the lexical and grammatical components extracted from word embeddings display greater internal consistency after distillation. In each comparison, we measure the target evaluation metric before and after the procedure. In all result tables, the *before* column reports results calculated using the original embeddings, while the *after* column shows results based on distilled vectors. The relative change (Δ) is calculated

as $\frac{after-before}{before}$, reflecting the proportion of the resulting change. All reported results are rounded to two decimals.

4.1 Lexical vectors become more consistent after grammatical removal

Table 2 reports the results of the evaluation of the lexical vector distilled by removing the grammatical component(s).

For each lexeme in our dataset, we select inflected words that differ in grammatical features and calculate the average pairwise L2 distance between their original embeddings. This distance serves as a measure of internal lexical dispersion prior to distillation. We use the same measure on lexical vectors derived after removing grammatical components. If the grammatical information has been successfully removed, the resulting lexical vectors should exhibit lower internal dispersion. We exclude lexemes with identical feature values, as their embeddings are already highly similar in the original space; the procedure would yield negligible effect.

As shown in Table 2, we observe a consistent decrease in distance across all lexical categories in the range of around 8% to 14%. This indicates that the distilled lexical vectors are more tightly clustered, supporting the effectiveness of our distillation method across different parts of speech.

For baseline comparison, we evaluate random word pairs drawn from different lexemes that differ in feature values. These words are not expected to share a semantic content, so removing grammatical information should not significantly reduce their distance. Indeed, the random groupings of the same part of speech exhibit notably smaller reductions, with average decreases reaching only about half of those observed in lexeme-aligned groups. This confirms that the increased compactness observed in structured lexeme groups reflects meaningful decomposition rather than trivial consequence of mean subtraction or vector manipulation.

4.2 Grammatical vectors get closer after distillation

In addition to lexical coherence, we also evaluate the internal consistency of the grammatical vectors, with results presented in Table 3.

For each feature value, we find words from distinct lexemes that share this value. The average pairwise L2 distance between their original embeddings measures how dispersed these words are before distillation. We then compute that distance using the grammatical vectors extracted after removing lexical components. If the subtraction is effective, these vectors should converge toward a representation of the shared grammatical property, lowering the average distance. Again, we omit tokens from the same lexeme to avoid trivial reductions stemming from shared lexical information.

As shown in Table 3, the grammatical vectors exhibit a substantial reduction in pairwise distance across all feature values within all grammatical categories, ranging from approximately 31% to 45%. This level of reduction is markedly higher than what we observed for lexical vectors. This stark contrast suggests that grammatical information is more effectively disentangled. A possible explanation lies in the structural difference between the comparison groups in each evaluation. In the lexical vector analysis, we compare tokens from the same lexeme that differ only in grammatical features. Such tokens already occupy relatively close positions in the embedding space even before distillation, leaving limited room for further convergence. By contrast, in the grammatical vector analysis, the compared tokens share a grammatical feature but are from different lexemes, are therefore initially more widely dispersed. After the lexical component is removed, this dispersion is greatly reduced, as the remaining grammatical vectors align more closely around the shared grammatical property. Moreover, grammatical features are often shared by a larger number of tokens than individual lexemes, making the averaged estimates for grammatical vectors more robust.

To establish a control, we measure distances between randomly sampled words that differ in both lexeme and feature value. Since such pairs are not expected to encode common grammatical information, their grammatical vectors should remain dispersed. This comparison ensures that the observed distance reductions in feature-based groups cannot be explained by vector subtraction alone. Table 4 shows reductions in distance consistently small across all grammatical categories, less than 10%. These values are markedly lower than those observed in structured feature-based groups (cf. Table 3), confirming that the substantial convergence seen reflects the extraction of meaningful shared grammatical information.

	Lexemes			Random lexemes		
	before	**after**	**Δ**	**before**	**after**	**Δ**
Noun	5.30	4.71	-10.32%	6.70	6.41	-4.24%
Adjective	4.72	4.34	-7.97%	6.78	6.50	-4.03%
Verb	5.66	4.82	-14.07%	7.14	6.48	-9.22%

Table 2: Pairwise distance of lexical vectors before and after distillation. Results are computed over word pairs from the same lexeme, as well as random word pairs from different lexemes, all inflected with different grammatical feature values.

	before	**after**	**Δ**
Noun			
sing	6.08	3.33	-45.18%
plur	7.34	4.23	-42.31%
Adjective			
sing	6.33	3.62	-42.85%
plur	7.15	4.35	-39.12%
masc	6.63	3.83	-42.19%
fem	6.54	3.88	-40.70%
Verb			
ind	7.09	4.40	-37.89%
per-3	7.02	4.36	-37.88%
sing	6.71	4.13	-38.50%
plur	7.57	4.70	-37.91%
pres	6.66	4.09	-38.51%
imp	7.81	4.77	-38.86%
inf	5.93	3.65	-38.36%
fin	7.06	4.81	-31.94%

Table 3: Pairwise distance of grammatical vectors before and after the distillation, calculated over word pairs that share the same grammatical feature value but originate from distinct lexemes.

	before	**after**	**Δ**
N-number	7.10	6.73	-5.20%
Adj-number	7.06	6.77	-4.14%
Adj-gender	6.67	6.55	-1.76%
V-mode	8.07	7.29	-9.66%
V-person	7.36	7.25	-1.42%
V-number	7.45	7.15	-4.08%
V-tense	7.75	7.34	-5.27%
V-finiteness	6.99	6.61	-5.42%

Table 4: Pairwise distance of random grammatical vectors before and after distillation, calculated between random word pairs differing in grammatical feature value and lexeme.

4.3 Word embedding reconstruction

Extending the above evaluations, to assess whether the distilled components provide more accurate representations of lexical and grammatical information, we evaluate how well they can reconstruct the original word embeddings. Specifically, we measure the L2 distance between the original embedding of each token and its reconstructed form, obtained by summing its lexical vector and the average of its grammatical vectors corresponding to each of its feature values.

	before	**after**	**Δ**
Noun	2.58	2.52	-2.38%
Adjective	2.89	2.58	-11.40%
Verb	3.50	2.27	-35.28%

Table 5: Reconstruction error of word embeddings from their lexical and grammatical components before and after the distillation. Results are averaged over part of speech.

As a baseline, we first perform reconstruction using the initial, undistilled estimates of lexical and grammatical vectors. These initial estimates are expected to contain overlapping or entangled information, resulting in higher reconstruction error. After distillation, however, the components are refined to better isolate the intended dimensions of meaning, which should lead to more faithful reconstructions.

Our results are reported in Table 5. We observe consistent reductions across all parts of speech. The improvement is most substantial for verbs, with an average reduction of over 35%, while adjectives and nouns show smaller but still meaningful improvements (11.4% and 2.4%, respectively). This pattern may be attributed to differences in morphological complexity and the way grammatical information is distributed across parts of speech. In French, both nouns and adjectives typically mark number and gender using the same surface morphemes (e.g., *-s* and *-x* for PLUR; *-e*,*-euse* and *-trice* for FEM), which are shared between lexemes. While such suffixes are consistent and formally simple, they express only a limited set of grammatical features, and the shared form across categories may blur the information, making it more difficult for the model to disentangle the lexical and grammatical components precisely.

In contrast, French verbs undergo more complex

inflection, where a single suffix often encodes multiple feature values simultaneously. For instance, the ending *-ent* in *ils parlent* ('they speak') marks third person, plural, present tense, and indicative mood all within a single affix. Despite the greater surface complexity, the richness and density of grammatical encoding in verbal morphology may provide a stronger signal, allowing the model to better isolate and represent grammatical content. The more pronounced improvement observed in verbs thus likely reflects this concentrated grammatical structure, which becomes more salient and recoverable after distillation.

5 Conclusion

We aimed to disentangle contextualized word embeddings in CamemBERT into lexical and grammatical parts. We proposed an iterative distillation method based on the complementarity of averaging and subtraction. A word's lexical vector can be approximated either by averaging the vectors of all words that share the same lexical meaning, or by subtracting the vectors corresponding to its grammatical features. Similarly, its grammatical vector can be obtained either by subtracting the lexical part from the original embedding, or by averaging the embeddings of all words that share the same grammatical feature.

If effectively separated, the lexical and grammatical vectors should be more distinct, with minimal overlap between their respective contents compared to their initial estimates. Each vector should convey more clearly the structural regularities shared with similar words, resulting in tighter alignment within their lexical or grammatical groups. As such, they are better suited to jointly approximate the original word embedding. As expected, in our evaluation, the final lexical and grammatical vectors that we extracted are more clearly clustered with their structurally similar counterparts, when combined, reproduce original word embeddings with minimal loss.

Notably, the reduction in distance is much more pronounced for grammatical vectors than for lexical vectors—around 40% versus 10% on average. Since the initial grammatical vectors are averaged over a large set of lexemes and contain great lexical noise, which is removed during distillation, leading to tighter alignment. The extent of this convergence is relatively stable across different feature values, but varies across parts of speech. Verbs, especially, show a stronger reduction in distance and a larger drop in reconstruction error compared to nouns and adjectives. This may be due to the richness of verbal morphology in French, where suffixes often encode several grammatical features at once, making the grammatical signal more prominent and its removal more effective. Nouns show only minor reconstruction gains, likely due to limited grammatical variation from number inflection alone. Also, large number of nouns in our dataset may stabilize their initial estimates, leaving less room for improvement. Adjectives fall in between, showing moderate gains.

Another important factor behind these observations concerns tokenization and the model's sensitivity to morphological markers. In CamemBERT, frequent, short, and morphologically informative tokens are more likely to be consistently encoded or even assigned special status during training (Rogers et al., 2021; Clark et al., 2019; Mohebbi et al., 2021). In contrast, lexical roots often span longer or rarer sub-word tokens and are more prone to being split or distorted, especially in low-frequency contexts. As a result, grammatical information is already more cleanly separated and clearly encoded in the model's internal representations, making it easier to distill effectively. This also explains why verbs, whose suffixes encode multiple features in compact forms, benefit the most from the process.

Future work will further explore how factors such as word frequency and tokenization affect the separation of lexical and grammatical vectors. Our method assumes a linear relationship between lexical and grammatical vectors; in a follow-up study, we plan to explore non-linear relationships. Given that our method does not rely on language-specific morphological rules, we will apply and evaluate it across languages. In addition, we are interested in extending the approach using learning-based methods, and in incorporating morphology-aware tokenizers to improve grammatical representation. Finally, we aim to assess the practical value of our decomposition through downstream tasks.

Limitations

One limitation of our study is data imbalance, which may affect result comparability and robustness. The number of words varies widely across parts of speech and grammatical features: nouns are far more numerous, yet have fewer grammatical features. Some feature values are sparsely rep-

resented, leading to less reliable vector estimates. Lexemes also vary in the number of inflected forms. Due to limited data, certain features such as verbal voice and gender were excluded, making the evaluation less complete.

Source code

This experiment can be reproduced by downloading the data we used and our source code from `https://github.com/liliulng/disentangle-wemb`.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable suggestions. We are also grateful for the financial support of the China Scholarship Council (#202008310177) and the Fonds de recherche du Québec (#366841).

References

ATILF. 2023. Réseau lexical du français (rl-fr). ORTOLANG (Open Resources and TOols for LANGuage)—www.ortolang.fr.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. A multi-task approach for disentangling syntax and semantics in sentence representations. *arXiv preprint arXiv:1904.01173*.

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 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL'19: Human Language Technologies*, volume 1, pages 4171–4186, Minneapolis, MN, USA. ACL.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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 55–65, Hong Kong, China. Association for Computational Linguistics.

James Y Huang, Kuan-Hao Huang, and Kai-Wei Chang. 2021. Disentangling semantics and syntax in sentence embeddings with pre-trained language models. *arXiv preprint arXiv:2104.05115*.

François Lareau, Gabriel Bernier-Colborne, and Patrick Drouin. 2015. La séparation des composantes lexicale et flexionnelle des vecteurs de mots. In *Actes de la 22e conférence sur le Traitement Automatique des Langues Naturelles. Articles courts*, pages 242–248, Caen, France. ATALA.

Dingcheng Li, Hongliang Fei, Shaogang Ren, and Ping Li. 2021. A deep decomposable model for disentangling syntax and semantics in sentence representation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4300–4310.

Li Liu and Francois Lareau. 2024. Assessing BERT's sensitivity to idiomaticity. In *Proceedings of the Joint Workshop on Multiword Expressions and Universal Dependencies (MWE-UD) @ LREC-COLING 2024*, pages 14–23.

Miguel López-Otal, Jorge Gracia, Jordi Bernad, Carlos Bobed, Lucía Pitarch-Ballesteros, and Emma Anglés-Herrero. 2025. Linguistic interpretability of transformer-based language models: a systematic review. *arXiv preprint arXiv:2504.08001*.

Veronika Lux-Pogodalla. 2014. Integrating lexicographic examples in a lexical network (intégration relationnelle des exemples lexicographiques dans un réseau lexical) [in French]. In *Proceedings of TALN 2014*, volume 2, pages 586–591, Marseille, France. ATALA.

Veronika Lux-Pogodalla and Alain Polguère. 2011. Construction of a French Lexical Network: Methodological Issues. In *First International Workshop on Lexical Resources (WoLeR 2011)*, pages 54–61, Ljubljana, Slovenia.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7203–7219, Online. ACL.

Igor A. Mel'čuk. 2006. Explanatory combinatorial dictionary. In Giandomenico Sica, editor, *Open Problems in Linguistics and Lexicography*, pages 225–355. Polimetrica, Monza, Italy.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems (NIPS 2013)*, 26.

Hosein Mohebbi, Ali Modarressi, and Mohammad Taher Pilehvar. 2021. Exploring the role of BERT token representations to explain sentence probing results. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 792–806.

Shiva Omrani Sabbaghi and Aylin Caliskan. 2022. Measuring gender bias in word embeddings of gendered languages requires disentangling grammatical gender signals. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pages 518–531.

Alain Polguère. 2009. Lexical systems: graph models of natural language lexicons. *Language Resources and Evaluation*, 43:41–55.

Alain Polguère. 2014. From writing dictionaries to weaving lexical networks. *International Journal of Lexicography*, 27(4):396–418.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Shauli Ravfogel, Yanai Elazar, Jacob Goldberger, and Yoav Goldberg. 2020. Unsupervised distillation of syntactic information from contextualized word representations. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 91–106, Online. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. A primer in bertology: What we know about how bert works. *Transactions of the association for computational linguistics*, 8:842–866.

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.

Kaitlyn Zhou, Kawin Ethayarajh, Dallas Card, and Dan Jurafsky. 2022. Problems with cosine as a measure of embedding similarity for high frequency words. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 401–423, Dublin, Ireland. Association for Computational Linguistics.

Author Index

A. Rodriguez, Maria, 185
Abzianidze, Lasha, 252
Alacam, Özge, 73
Anand, Gunjan, 317
Asher, Nicholas, 176

Baggio, Giosuè, 88
Blodgett, Austin, 16
Bonial, Claire, 16
Boritchev, Maria, 40
Bos, Johan, 59, 147
Brutti, Richard, 292

Candito, Marie, 185
Chaturvedi, Akshay, 176
Chersoni, Emmanuele, 88, 218
Coavoux, Maximin, 40
Cooper, Robin, 128

Dalrymple, Mary, 199
Dobnik, Simon, 299
Dominguès, Dominguès, 241
Donatelli, Lucia, 292
Dunn, Jonathan, 317

Ehren, Rafael, 166
Evang, Kilian, 166

Feng, Zhaoxin, 218
Ferraro, Francis, 16
Forkel, Robert, 1
Frassinelli, Diego, 279

Ginzburg, Jonathan, 128
Grant, Ian Paul, 153
Guembour, Sami, 241
Guiomar, Gonçalo, 308

Hoeken, Sanne, 73
Huang, Chu-Ren, 218
Huyghe, Richard, 185

Jamil, Huma, 51
Jang, Hyewon, 279
Jie, LU, 99
Jin, Du, 99

Kallmeyer, Laura, 166
Kang, Jeongwoo, 40
Khebour, Ibrahim, 51
Kozachenko, Ekaterina, 308
Krishnaswamy, Nikhil, 51
Kulcsar, Margareta A., 153
Künkele, Dominik, 299

Lai, Kenneth, 51, 292
Langlais, Philippe, 108
Lareau, François, 108, 331
Larsson, Staffan, 128
Lenci, Alessandro, 88
LI, Shuxu, 108
List, Johann-Mattis, 1
Liu, Dong, 30
Liu, Li, 331
Lücking, Andy, 128

Matsuoka, Daiki, 137
Meng, Qianru, 59
Mikami, Yosuke, 137
Moot, Richard, 252

Nguyen, Dong, 73

Obiso, Timothy, 264

Patejuk, Agnieszka, 199
Pellegrin, Taylor, 16
Petersen, Wiebke, 120
Ploux, Sabine, 241
Poesio, Massimo, 73, 153
Pustejovsky, James, 51, 264, 292

Radaelli, Matteo, 88
Retoré, Christian, 252
Robillard, Simon, 252
Rudinger, Rachel, 16
Rzymski, Christoph, 1

Shichman, Mollie, 16
Skandalis, Maximos, 252
Song, Huacheng, 218
Spalek, Katharina, 120
Stanczak, Karolina, 308

Thompson, Kate, 176
Tjuka, Annika, 1

Tu, Jingxuan, 264

Venant, Antoine, 108

Yanaka, Hitomi, 99, 137
Ye, Bingyang, 264
Yu, Yanxuan, 30

Zarrieß, Sina, 73
Zhang, Xiao, 59, 147
Zymla, Mark-Matthias, 199