BriGap 2026

Third Workshop on the Bridges and Gaps between Formal and Computational Linguistics

Proceedings of the Workshop

July 11, 2026

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

Founded by

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

Introduction

We are excited to welcome you to BriGap-3 in Paris, France!

This third edition of the workshop on Bridges and Gaps between Formal and Computational Linguistics is the first to be organised independently of any other conference. It follows up on the first edition in 2022 (co-located with ESSLLI) and the second in 2025 (co-located with IWCS). We have decided to continue with the dual submission policy implemented for the second edition, soliciting archival papers to be published in the ACL Anthology, and non-archival submissions describing work in progress or work published elsewhere that would be of interest to our audience.

We are very pleased with the response to the call for this third edition. Out of the submissions received, 21 were selected for presentation (8 oral presentations and 13 poster presentations), of which 13 are included in these proceedings. These contributions sit at the intersection of theoretical linguistics and computational linguistics or natural language processing, spanning a wide range of topics such as formal semantics and inference, language models and linguistic theory, pragmatics and common ground, grammar engineering and low-resource languages, and human–model comparison.

In addition to these contributed presentations, BriGap-3 includes two invited talks. Raquel Fernández (Universiteit van Amsterdam) will explore how machine learning models trained on body movements, vision, and language can shed light on multimodality in human cognition, whereas Tal Linzen (New York University) will show how formal symbolic models can be leveraged to make LLM pretraining more efficient, evaluate their capabilities, and improve how they update their beliefs during interaction. Together, the two keynotes bridge cognitively and formally grounded approaches to language, setting the stage for a rich exchange between formal and computational linguistics.

BriGap-3 was made possible thanks to the financial support of RT LIFT2, a France-based research group aiming to bring together researchers in computational linguistics, formal linguistics, and field linguistics around shared questions, data, and tools, LLF, a joint Université Paris Cité-CNRS laboratory doing research covering a wide range of topics in formal, experimental and computational linguistics, and InIdEx EFL, an Université Paris Cité project that supports innovative research on languages and language, bringing together different fields of linguistics and related disciplines (psychology, neuroscience, computer science) around themes that respond to the major challenges of today's world.

Timothée Bernard, Giulia Rambelli, Emmanuele Chersoni, Program Chairs

Program Committee

Senior Program Committee

Timothée Bernard, Université Paris Cité, CNRS, Laboratoire de linguistique formelle
Emmanuele Chersoni, Hong Kong Polytechnic University
Giulia Rambelli, University of Aix-Marseille, CNRS, Laboratoire Parole et Langage / Laboratoire d'Informatique et Systèmes

Program Committee

Lasha Abzianidze, Utrecht University
Pascal Amsili, Sorbonne Nouvelle (Paris 3)
Giosuè Baggio, Norwegian University of Science and Technology
Sacha Beniamine, University of Surrey
Philippe Blache, Aix-Marseille University, CNRS
Johan Bos, University of Groningen
Chloe Braud, IRIT, CNRS, ANITI
Canaan Breiss, University of Chicago
Marie Candito, Université Paris Cité
Yan Cong, Purdue University
Benoît Crabbé, Université Paris Cité
Luca Dini, Institute of Computational Linguistics (CNR, Pisa)
Abdellah Fourtassi, Aix-Marseille University
Yu-Yin Hsu, Hong Kong Polytechnic University
Gene Kim, University of South Florida
Dan Lassiter, University of Edinburgh
Timothèe Mickus, University of Helsinki
Koji Mineshima, Keio University
Filip Miletic, University of Stuttgart
Ludovica Pannitto, University of Bologna
Chiara Paolini, KU Leuven
Denis Paperno, Utrecht University
Vered Shwartz, University of British Columbia
Ece Takmaz, Utrecht University
Kees Van Deemter, Utrecht University
Grégoire Winterstein, Université du Québec à Montréal
Guillaume Wisniewski, Université Paris Cité
Olga Zamaraeva, University of Helsinki
He Zhou, Hong Kong Polytechnic University

Keynote Talk

Modelling Multimodality in Human Cognition with Machine Learning Models

Raquel Fernández
Universiteit van Amsterdam

Abstract: Linguistic communication is both inherently multimodal and interactive. Conceptual knowledge encoded in language is grounded in our sensory-motor experience, while in face-to-face dialogue we use speech in tandem with non-verbal signals such as gaze and gestures. In this talk, I will present our work on using machine learning models trained on body movements, vision, and language to study fundamental questions regarding multimodality in human cognition. I will first focus on behaviour: I will discuss our approach to co-speech gesture representation learning, showing that the resulting gesture embeddings exhibit properties that support theoretically motivated hypotheses. I will then move to describing our experiments on modelling brain activity with pre-trained vision-language models, which add to existing evidence for the intricate relationship between language and perception in the human brain.

Keynote Talk

Formal symbolic models for LLMs: pretraining, evaluation and post-training

Tal Linzen
New York University

Abstract: Formal symbolic models—ideal versions of the computational problems involved in learning about and interacting with the world—are central in linguistics and cognitive science. These models make it possible to generate unlimited amounts of synthetic data of variable complexity, as well as define verifiably correct outcomes for each instance of the problem. I will discuss three studies that leverage these properties of formal models. First, I will show that by pretraining transformer LLMs on formal languages before training them on natural languages, we can make training more compute and data efficient overall. Second, I will introduce context-free language recognition as an evaluation task for LLM. I will show that the complexity of the grammar reliably predicts the model's accuracy on this task, and that even the strongest reasoning models available struggle to perform this task as the complexity of the language increases. Finally, I will demonstrate how symbolic Bayesian models can be used to evaluate and improve LLM abilities to update their probabilistic beliefs when interacting with users.

Table of Contents

From Execution to Exploration: Bridging the Usability Gap in Formal Natural Language Inference

Koharu Saeki
Ochanomizu University
saeki.koharu@is.ocha.ac.jp

Daisuke Bekki
Ochanomizu University
bekki@is.ocha.ac.jp

Abstract

Linguistically oriented formal NLI systems ensure the validity and transparency of inference. However, the combinatorial explosion of candidates, which we term the *branching problem*, imposes prohibitive computational overhead and a heavy cognitive burden on grammar developers. We argue that a central cause is a mismatch between the exhaustive execution paradigm and the actual workflow of grammar developers. To overcome this barrier, we propose restructuring the development workflow from exhaustive execution to interactive exploration driven by developer decisions. We realize this shift in *Express*, a web-based interactive development environment for *lightblue*, a Japanese automated inference system built upon Combinatory Categorial Grammar and Dependent Type Semantics. *Express* transforms branches at each stage of parsing, type checking, and proof search into explicitly selectable units, transferring control over the reasoning process to the developer. Our evaluation shows that this paradigm shift effectively reduces unnecessary computation and cognitive burden during grammar development: in a user study, we observed a 96% reduction in explored paths and an improvement in the task success rate from 25% to 100%. Furthermore, a case study demonstrates a roughly 12× reduction in debugging turnaround time.

1 Introduction

Linguistically oriented approaches to natural language inference (NLI), ranging from automated systems (Bos, 2008; Mineshima et al., 2015; Abzianidze, 2017; Hu et al., 2019) to proof-assistant-based approaches (Chatzikyriakidis and Luo, 2014), ground inference in logical forms and theorem proving, ensuring both validity and transparency. Yet the grammar developers who build and maintain these systems struggle to use them effectively in everyday development.

The system targeted in this work, *lightblue* (Tomita et al., 2025), is a formal NLI system based on CCG and DTS (see §2.1 for details). *lightblue* serves two complementary use cases: as an automated inference engine whose outputs are used for NLI experiments and treebank construction (Tomita et al., 2024), and as a platform for grammar development, where developers trace through the inference pipeline to verify that their theoretical assumptions yield the intended linguistic predictions. It is this second use case—a feedback loop between linguistic theory refinement and computational verification—that *Express* is designed to support.

However, the development and verification of such systems pose unique challenges. Inference in *lightblue* consists of multiple stages (syntactic/semantic parsing, type checking, and proof search), each of which retains multiple theoretically admissible candidates as it proceeds. We refer to the resulting proliferation of candidates as the *branching problem*. Unlike conventional NLI systems where sentence-level ambiguities are resolved independently, *lightblue*'s DTS-based architecture performs type checking sequentially: each syntactic structure may yield multiple type checking outcomes due to alternative anaphora/presupposition resolutions, and each such outcome serves as the context for type checking the subsequent sentence. As a result, ambiguities compound across sentences, causing the total number of branches to grow exponentially even for a typical NLI problem with just two or three input sentences.

Under the conventional exhaustive execution approach, computation is uniformly applied to all branches, including those irrelevant to the developer's interests. In practice, developers must manually sift through voluminous output to locate a single path of interest among hundreds of candidates, a cognitively demanding and error-prone process that constitutes a major bottleneck in everyday

Third Workshop on the Bridges and Gaps between Formal and Computational Linguistics, pages 1–11

grammar development.

While interactive tools exist for grammar engineering and NLP pipeline inspection—XLE (Crouch et al., 2011) for LFG, INCEpTION (Klie et al., 2018) for annotation, AllenNLP Demo (Gardner et al., 2018) for visualization—they target a single processing phase and do not track chains of ambiguity across fundamentally different stages such as parsing, type checking, and theorem proving.

More closely related is the use of proof assistants for NLI: Chatzikyriakidis and Luo (2014) use Coq's interactive capabilities to verify natural language inferences. However, proof assistants operate within a single proof goal, whereas the branching problem targeted here arises upstream, at the level of selecting which syntactic structures and type checking outcomes to pass to the prover. Our approach makes this upstream disambiguation itself interactive, a level of control that proof assistants do not address.

To our knowledge, no integrated interactive development environment has been proposed for formal NLI systems that provides unified visualization and control of ambiguity across parsing, type checking, and proof search.

We argue that the root cause of this usability issue lies not in computational complexity or interface design, but in a fundamental gap between the exhaustive execution paradigm (which computes everything before presenting results) and the actual workflow of grammar developers (who naturally focus on specific branches, verify them incrementally, and apply their linguistic intuition to select among candidates).

To bridge this gap, we propose restructuring the development workflow from execution to exploration: instead of inspecting exhaustively computed results, developers navigate the inference pipeline interactively, guided by their linguistic judgment. We realize this shift in *Express*, a web-based interactive development environment for *lightblue* that makes branches explicitly selectable units and transfers control over the reasoning process back to the developer. Implemented in Haskell—the same language as *lightblue*—using the Yesod framework (Snoyman, 2015), *Express* directly reuses *lightblue*'s internal data structures.

Our contributions are threefold: (1) An interactive execution model for *lightblue* in which disambiguation advances incrementally through user interaction; (2) *Express*, a web-based environment with unified branch visualization across parsing, type checking, and proof search; and (3) Empirical validation through a user study demonstrating substantial improvements in both task success rate and completion time.

Our evaluation shows that this shift effectively avoids unnecessary computation and reduces cognitive burden (§5).

2 The Branching Problem

This section formally defines how branches proliferate across *lightblue*'s multi-stage inference pipeline (syntactic/semantic parsing, type checking, and proof search), and argues that the resulting computational overhead and cognitive burden stem from a paradigm mismatch.

2.1 Multi-Stage Inference in Formal NLI

lightblue is an automated inference system that integrates Combinatory Categorial Grammar (CCG; Steedman, 2000) with Dependent Type Semantics (DTS; Bekki and Mineshima, 2017). Its inference pipeline consists of three sequential stages (Tomita et al., 2025):

Parsing and semantic composition. CCG-based syntactic parsing is applied to input sentences, yielding DTS-based semantic representations. Since CCG parsing employs CYK chart parsing, multiple grammatically admissible syntactic structures may arise from various combinations of syntactic types or alternative category assignments to lexical items.

Type checking and anaphora resolution. Type checking under Dependent Type Theory (Martin-Löf, 1984) is applied to each resulting semantic representation. When a representation contains underspecified terms (e.g., pronouns and other presupposition triggers that require resolution), anaphora resolution is simultaneously carried out through proof search; this process may yield multiple proof terms, causing the type checking results to branch. For multi-sentence inputs, the semantic representation of each preceding sentence serves as the context for type checking the subsequent one, causing the combinations of syntactic structures and type checking outcomes to grow exponentially in the number of sentences.

Proof search. The theorem prover *wani* (Daido and Bekki, 2020) determines whether a proof term can be constructed for entailment (the hypothesis sentence) or contradiction (its negation) to output YES, NO, or UNKNOWN labels.

2.2 Formal Definition of the Branching Problem

Given k input sentences ($k-1$ premises and one hypothesis), let i denote the maximum number of syntactic structures retained per sentence and j the maximum number of type checking outcomes per structure. The total number of proof search queries grows exponentially in k, reaching:

$$2 \cdot (i \cdot j)^k \tag{1}$$

In practice, parsing may yield multiple syntactic structures from syntactic type ambiguity, but only the top-n are retained; similarly, each structure may contain underspecified terms whose anaphora resolution and presupposition binding admit multiple proof terms, but only the top-m type checking outcomes are kept. Thus i and j in Equation (1) denote these bounded counts (e.g. $i = 4$, $j = 4$ in §5), yet the branching problem persists: because type checking proceeds sequentially, each sentence's result serving as context for the next, the total still grows as $(i \cdot j)^k$, exponentially in the number of sentences k.

We term this the *branching problem* (Figure 1). The factor of 2 reflects proof search for both entailment and contradiction.

Under the conventional exhaustive execution approach, all candidates are enumerated and processed uniformly. This leads to two critical consequences:

Computational overhead. In this paradigm, all potential paths are computed indiscriminately, including those irrelevant to the developer's linguistic intent. Even for typical NLI problems consisting of only two or three sentences, the number of branches increases exponentially with respect to k, requiring substantial computational resources to isolate the inference result for a specific path of interest.

Cognitive burden. The exhaustive approach intermingles results from all disambiguation paths, often flooding the developer with outputs from spurious or unintended branches. Developers are forced to manually sift through voluminous, interleaved data to locate the single reasoning path they wish to verify—a cognitively demanding and error-prone process that stifles rapid, iterative development and debugging.

2.3 The Root Cause: A Paradigm Mismatch

The branching problem reflects a fundamental mismatch between the system's "compute-then-output" execution paradigm and the actual workflow of grammar developers, who rarely need all inference results simultaneously.

For instance, a developer may wish to isolate a specific syntactic structure for a single sentence to observe its downstream effects on type checking and proof search. Alternatively, they may need to verify whether a recent modification to a grammar correctly reflects in the intermediate syntactic structures without running the entire pipeline. In these scenarios, what the developers actually require is selective verification, step-by-step control, and inspection of intermediate stages.

The exhaustive execution paradigm fundamentally fails to support this selective, incremental workflow. Because results are exclusively presented after the entire computation completes, the system operates as a monolithic process. The developer cannot intervene at intermediate stages of inference to prune unintended branches, directly causing the computational waste and cognitive overload discussed previously.

3 From Execution to Exploration

The mismatch identified in the previous section is rooted in the execution paradigm itself. Importantly, the shift we propose does not alter *lightblue*'s automated inference engine; rather, it restructures the development workflow so that developers can navigate the inference pipeline interactively, as detailed below.

Under the exploration paradigm we propose, developers navigate the inference pipeline stage by stage: at each stage, the system presents the available branch candidates, and the developer selects one before computation proceeds. Because the developer explicitly selects one candidate at each branching point, the number of active paths never grows to $2 \cdot (i \cdot j)^k$ as under exhaustive execution, but remains manageable at all times, simultaneously avoiding unnecessary computation and reducing cognitive burden.

This shift is not merely an interface improvement: a post-hoc filtering approach that layers a

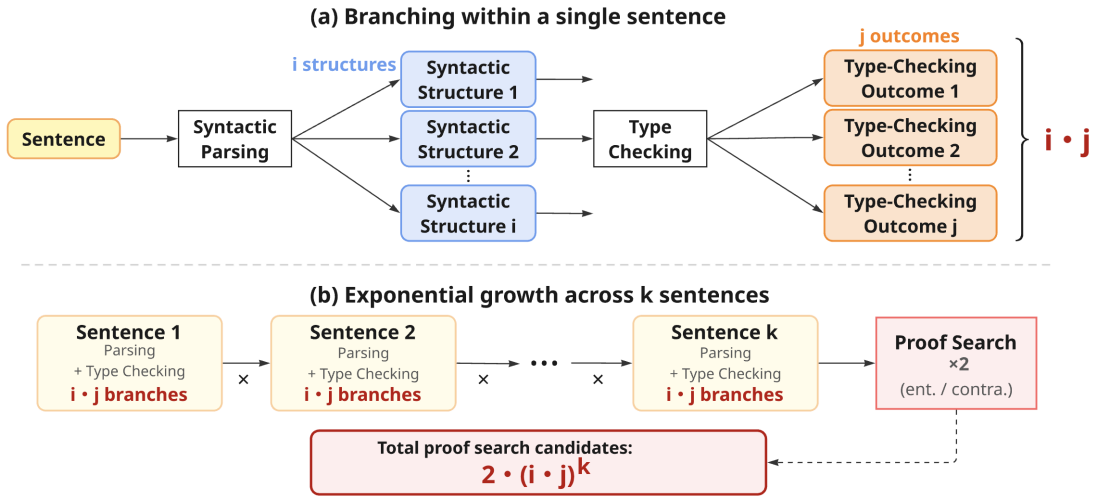


Figure 1: Branching structure of inference in *lightblue*. Each sentence retains i top-scoring syntactic structures and j type checking outcomes per structure, resulting in up to $2 \cdot (i \cdot j)^k$ proof search queries for k input sentences.

tree-structured UI over exhaustively computed results would reduce cognitive burden but would entirely fail to address computational overhead. Indeed, when relevant branches are not reached within practical time limits (as with Problem B in §5.1), there are no results to filter. The exploration paradigm resolves this by integrating developer decisions into the execution itself: exploiting Haskell's lazy evaluation (§4.3), computation proceeds on demand only along the path the developer selects, so that results can be obtained even when exhaustive enumeration is infeasible.

4 Express

Express realizes the exploration paradigm as a web-based interactive development environment for *lightblue*, enabling step-by-step control over branch selection across the entire inference pipeline.[1]

4.1 Design Principle

The design of *Express* is guided by the principle that navigating the inference pipeline should be an interactive, user-driven process rather than a monolithic batch computation. In this architecture, branches are treated as explicitly selectable units that the developer can inspect and operate on. Specifically, syntactic structures at the parsing stage and type checking outcomes at the semantic stage are presented as distinct candidates for the developer to evaluate. While the current implementation directly interfaces with *lightblue*'s internal functions, the exploration logic itself is not inherently system-specific. Abstracting the parser and type checker interaction into a Haskell type class, with *lightblue* as one instance, is a planned extension to support other formal NLI systems.

Based on this principle, *Express* integrates three features aligned with the practical debugging workflow: (1) **Investigate:** localized substring analysis; (2) **Explore:** interactive branch selection for full-sentence inference; and (3) **Share:** proof search query export for cross-developer collaboration. Each is detailed in the following subsections.

4.2 Investigate: Substring Analysis

The first step in diagnosing unexpected behavior is to localize the problem to a specific substring. *Express* builds on *lightblue*'s CYK chart parsing—which inherently retains syntactic structures for all intermediate constituents—allowing the developer to select any arbitrary span of the input sentence. The system instantly displays the parsing candidates and their DTS semantic representations for that span (Figure 2), enabling rapid localized diagnosis without re-running the full pipeline.

4.3 Explore: Interactive Branch Selection

Building on this local analysis, the developer must then run full-sentence inference to verify how local modifications affect the final outcome. Under exhaustive execution, the developer would be forced

[1] Source code: https://github.com/kohapizza/lightblue (*Express* is integrated into *lightblue*). A demonstration and usage instructions are available at https://kohapizza.github.io/Express-Demo-Website/.

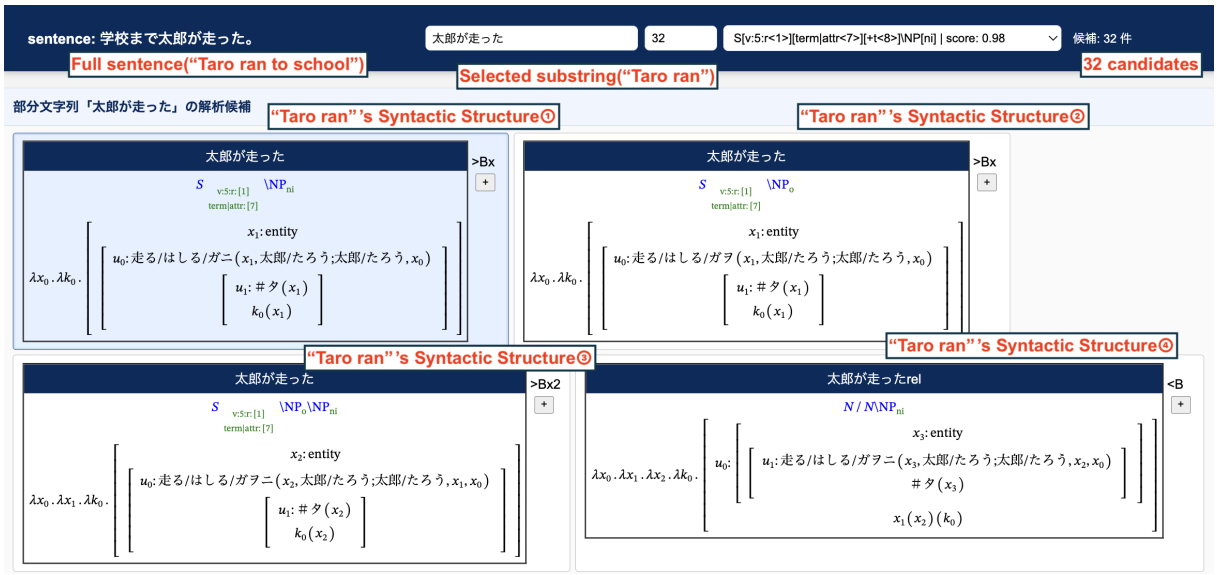


Figure 2: Substring analysis in *Express*. From the full sentence *Taro ran to school*, the developer selects the substring *Taro ran*. *Express* instantly displays all parsing candidates retained in the chart for the selected substring, each showing the CCG syntactic type and DTS semantic representation.

to wade through up to $2 \cdot (i \cdot j)^k$ paths to locate the relevant one.

Express eliminates this bottleneck by decomposing the inference process into two sequential selection points: (1) *syntactic structure selection:* Multiple parsing candidates generated by CCG are displayed as parallel, collapsible cards. The developer selects a single structure to proceed to type checking. (2) *Type checking outcome selection:* Multiple proof terms arising from alternative anaphora/presupposition resolutions are presented as distinct branches. The developer selects one specific outcome to proceed to proof search. Because the developer selects exactly one candidate at each stage, the number of active paths never grows exponentially (Figure 3 and Figure 4).

To ensure responsiveness, *Express* computes inference results on demand via lazy streams. Parsing results are maintained as per-sentence streams, and type checking outcomes are stored in a map keyed by sentence and node indices; changing a branch invalidates only downstream computations. Proof search results are cached by query content, so that different paths converging on the same query share cached results, minimizing redundant computation.

4.4 Share: Proof Search Query Export

When interactive exploration reveals a failure during the proof search phase, the grammar developer must communicate the failure conditions to the theorem prover’s developer. Historically, reproducing specific inference contexts required manually reconstructing the formal proof search query from raw system outputs, a tedious, error-prone process often consuming upwards of an hour for complex semantic representations.

Express directly addresses this friction by providing a feature to export the interactively formulated proof search query into a format natively loadable in *wani*. The theorem prover developer can then instantly load the exported query to accurately reproduce the failure context. This direct handoff accelerates the debugging cycle from initial diagnosis to collaborative resolution (Figure 5).

5 Evaluation

We now examine whether the shift from execution to exploration, as realized in *Express*, reduces unnecessary computation and cognitive burden caused by the branching problem (§2).

5.1 Search Space Reduction

We quantitatively evaluated the search space reduction using two NLI problems that represent typical scenarios grammar developers face: Problem A, which *lightblue* solves correctly but where exhaustive execution is slow due to branching; and Problem B, which is theoretically solvable but not handled successfully by the current theorem prover. Both problems consist of two input sentences ($k = 2$), with the system configured to retain up to 4 syntactic structures per sentence ($i = 4$; see

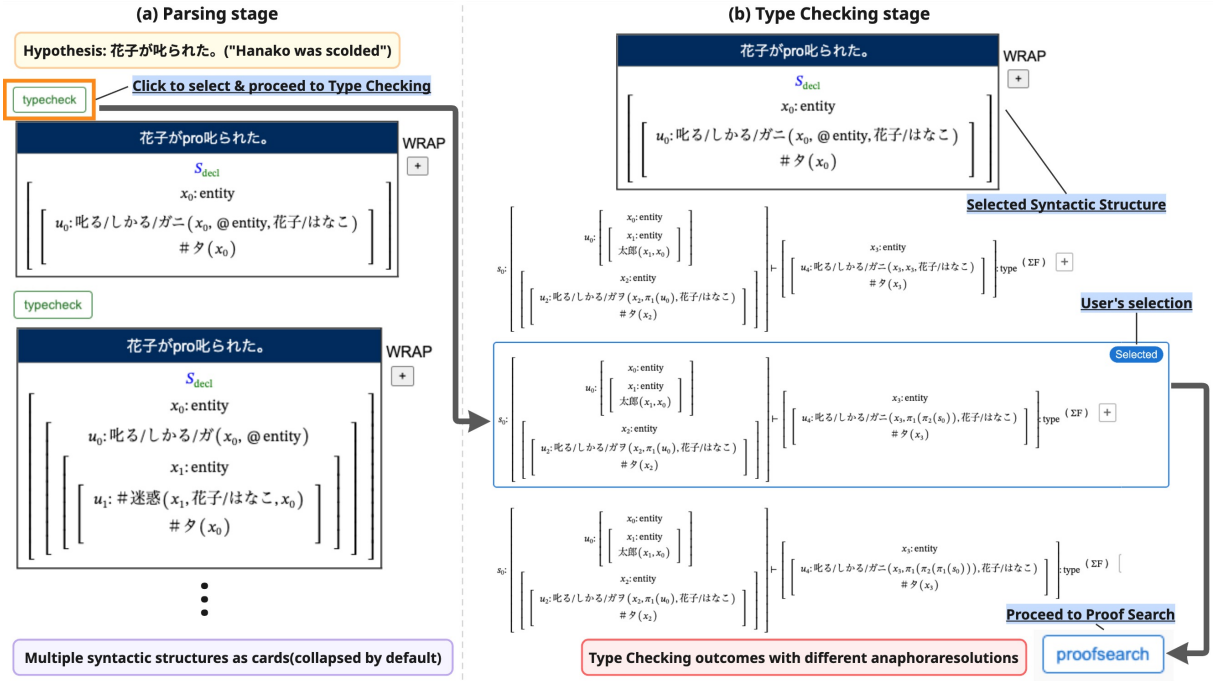


Figure 3: Interactive branch selection in *Express* for the hypothesis *Hanako was scolded*. (a) Multiple syntactic structures are displayed as collapsible cards (UI panels); the developer selects one by clicking `typecheck`. (b) The resulting type checking outcomes, each corresponding to a different anaphora resolution, are displayed; the developer selects one before proceeding to proof search.

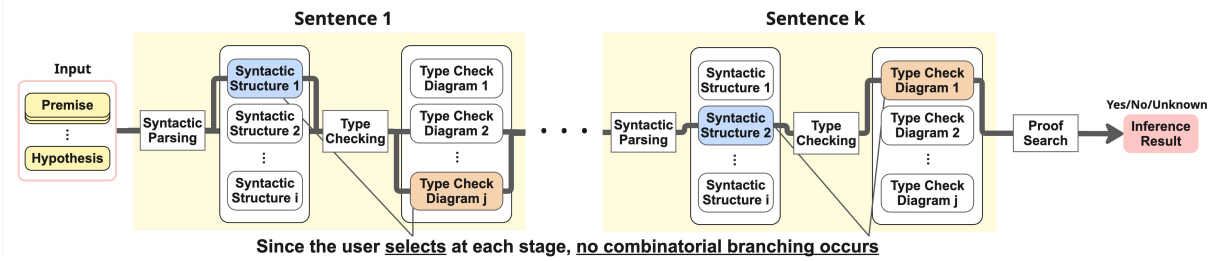


Figure 4: Operational flow of *Express*. The developer incrementally selects syntactic structures and type checking results at each stage (highlighted path), preventing the exponential branching that arises under exhaustive execution.

	Exhaustive		*Express*	
	Paths	Time	Paths	Time
Problem A	260	>120s	10	12s
Problem B	128	>120s	28	94s

Table 1: Search space comparison. Exhaustive execution was terminated at the 120-second time limit. Paths: number of disambiguation paths computed; Time: backend computation time.

§2.2) and up to 4 type checking outcomes per structure ($j = 4$), producing at most $2 \cdot (4 \cdot 4)^2 = 512$ proof search queries under exhaustive execution.

For Problem A, exhaustive execution generated 260 paths and timed out, whereas *Express* required only 10 paths and completed in 12 seconds, a 96% reduction (Table 1). For Problem B, exhaustive execution had evaluated 128 paths when terminated at the 120-second time limit; the enumeration remained incomplete and did not reach the target proof search query, which exists in the full search space. In contrast, *Express* enabled the developer to reach this query after evaluating only 28 paths and export it, enabling the kind of collaborative debugging illustrated in §5.3. These results show that interactive exploration effectively avoids unnecessary computation: developers can reach diagnostically relevant branches that are theoretically present but practically inaccessible under exhaustive execution.

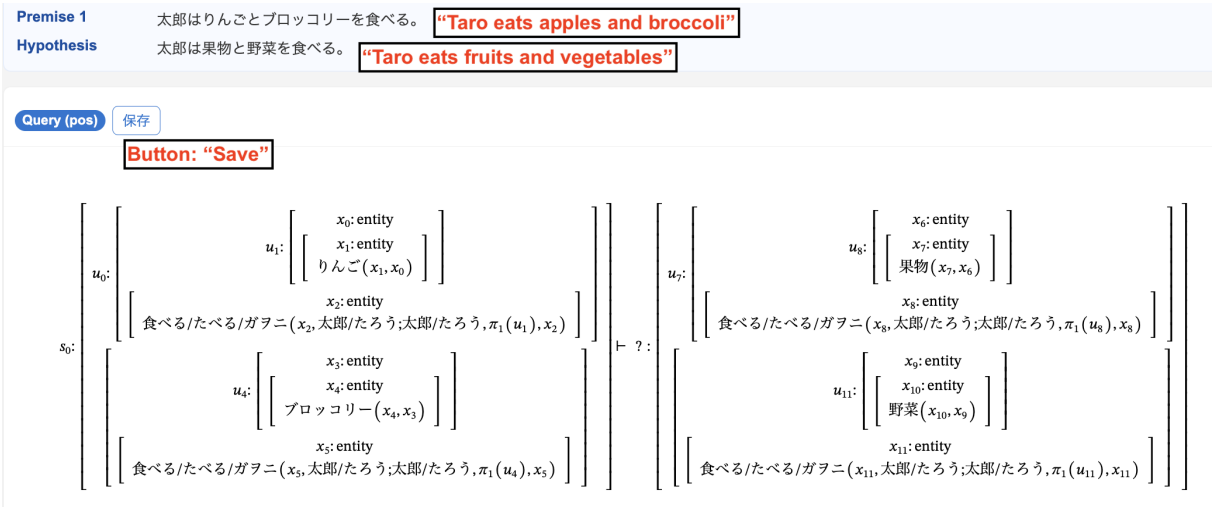


Figure 5: Proof search query export in *Express*. The premise *Taro eats apples and broccoli* and hypothesis *Taro eats fruits and vegetables* are displayed with the corresponding proof search query as a DTS type judgment. The query can be saved as a file directly loadable by the theorem prover *wani*.

5.2 User Study

To examine whether the reduction in search space translates into a reduced cognitive burden for developers, we conducted a within-subjects user study.

Participants and task design. Four researchers with experience in grammar development using *lightblue* participated, representing nearly the entire population of active *lightblue* grammar developers. The task was to identify a specific disambiguation path and locate the corresponding proof search query, using Problems A and B from §5.1. In the *Express* condition, participants used *Express*; in the baseline, they navigated a static HTML file containing all disambiguation paths from exhaustive execution. To control for order effects and problem difficulty, we employed a counterbalanced design: each participant solved one problem with *Express* and the other with the baseline, varying both the problem assignment and presentation order across participants (Table 2). This design directly isolates and measures the usability of inference process navigation, decoupled from *lightblue*'s underlying inference accuracy.

Results. A notable result is the contrast in task success rates: all four participants successfully identified the target path in the *Express* condition (100%), whereas only one succeeded in the baseline condition (25%).

The mean completion time decreased from 7m 20s to 1m 51s (Table 2). Note that the baseline mean is highly conservative; two participants reached the 10-minute time limit without completing the task, meaning their true completion times would have been longer. Participants' qualitative feedback indicates that the improvement lies not merely in speed, but in reduced cognitive load. In the baseline condition, P2 reported "I couldn't tell where I was in the file, and couldn't return to where I had been looking." In contrast, *Express* provided "an intuitive sense of the path." This suggests that transferring control over the reasoning process to developers enables them to maintain their spatial orientation within the inference space, substantially reducing the cognitive burden imposed by the exhaustive paradigm.

ID	Cond.	Prob.	Time	Ops.	Success
P1	Baseline	A	4m 08s	46	Yes
P1	*Express*	B	1m 26s	8	Yes
P2	Baseline	B	10m 00s	–	No (timeout)
P2	*Express*	A	1m 45s	14	Yes
P3	*Express*	A	2m 55s	16	Yes
P3	Baseline	B	10m 00s	–	No (timeout)
P4	*Express*	B	1m 17s	8	Yes
P4	Baseline	A	5m 12s	–	No
Baseline			7m 20s (avg)	46	1/4 (25%)
Express			1m 51s (avg)	11.5 (avg)	4/4 (100%)

Table 2: User study results. Rows are ordered by presentation sequence within each participant. Prob.: problem from Table 1. Time: task completion time (timeout at 10 min). Ops.: number of operations; omitted for failed trials.

Subjective usability ratings (Table 3) corroborate this interpretation: *Express* received near-maximum scores across all items, with a particularly pronounced advantage in the ease of under-

	Q1	Q2	Q3	Q4
Baseline	1.75 (1.5)	1.25 (1.0)	–	–
Express	4.75 (5.0)	4.75 (5.0)	5.00 (5.0)	4.75 (5.0)

Table 3: Likert scale results: mean (median). Q1: ease of reaching the target path; Q2: ease of understanding the inference process; Q3: workload reduction; Q4: intention to use in practice.

standing the inference process (Q2: median 5.0 vs. 1.0).

5.3 Case Study: Diagnosing Proof Search Failures

Finally, we illustrate how the exploration paradigm fosters not only individual efficiency but also collaborative problem-solving across distinct developer roles.

Given the premise "Taro eats apples and broccoli" and the hypothesis "Taro eats fruits and vegetables," *lightblue* returned an UNKNOWN label instead of the expected YES. This entailment should hold once proof search establishes the lexical relations—that apples are a kind of fruit, and broccoli a kind of vegetable. Diagnosing why it did not requires isolating, among the up to $2 \cdot (i \cdot j)^k$ candidates produced under exhaustive execution, the proof search query at issue. The grammar developer carried out this diagnosis in three steps, each corresponding to one of the features in §4.

Localizing the problem (*Investigate*). The developer first employed substring analysis (§4.2) to verify that the relevant segments of the input sentences were syntactically and semantically parsed as intended, thereby isolating the issue and confirming that it did not originate in the syntactic analysis phase.

Pinpointing the unsolved query (*Explore*). The developer then ran full-sentence inference and, at each branching point, selected the intended syntactic structure and type checking outcome (§4.3). Without sifting through the candidates produced by exhaustive execution, this led to the proof search query that should have returned YES but instead returned UNKNOWN. Since the query remains unsolved despite these intended selections, the problem can be attributed to proof search rather than to any earlier stage.

Reporting it to the prover developer (*Share*). Finally, the developer exported this specific query as a file natively loadable by *wani* (Figure 5) and sent it to the theorem prover's developer. Upon loading the file, the *wani* developer reproduced the exact state and identified the root cause, a search efficiency issue related to equality type introduction, within approximately five minutes.

Under the previous workflow, the *wani* developer would have had to manually transcribe the formal expressions from screenshots into DTT format, a tedious process that alone required roughly an hour for complex semantic representations. *Express*'s export feature entirely eliminated this transcription bottleneck, resulting in a roughly $12\times$ reduction in debugging turnaround time.

This case demonstrates that the exploration paradigm extends beyond individual productivity: by enabling developers to isolate and share specific branches in a reproducible format, *Express* makes cross-stage collaboration practically viable.

6 Conclusion

To address the computational overhead and cognitive burden caused by the branching problem, rooted in a mismatch between the exhaustive execution paradigm and the iterative workflow of grammar developers, we proposed restructuring the development workflow from execution to exploration. *Express*, the system realizing this paradigm, transforms branches into explicitly selectable units, transferring control over the reasoning process to the developer. As confirmed by our evaluation (§5), this shift substantially reduces unnecessary computation and cognitive burden in practice.

While our evaluation is currently limited to *lightblue* and Japanese data, we emphasize that its goal is not broad user generalization, but to assess usability for expert developers engaged in grammar engineering.

A natural concern is whether these gains reflect differences in task familiarity or problem difficulty rather than the tool itself. Our within-subjects, counterbalanced design guards against this: each participant performed one task with *Express* and the other with the baseline, with problem assignment and presentation order counterbalanced across participants (Table 2), controlling for both individual expertise and problem difficulty. The effect is moreover consistent across participants: all four succeeded with *Express*, whereas only one succeeded with the baseline and two trials reached the 10-minute timeout. We nonetheless acknowledge that $N = 4$ limits statistical power. Because

this already constitutes nearly the entire population of active *lightblue* developers, a substantially larger study would require that community to grow, which is itself one of the motivations for building *Express*.

While *Express* currently targets *lightblue*, the branching problem is not a limitation of this particular system but an inherent consequence of preserving syntactic ambiguity through the pipeline and performing anaphora resolution within cross-sentential type checking, a design that faithfully reflects the underlying linguistic theory. This combination is uncommon among current formal NLI systems, where ambiguity is typically resolved at a single stage (see Appendix A.1), and the branching problem accordingly does not arise. The paradigm shift we propose, from exhaustive execution to interactive exploration, is not specific to *lightblue*; it can apply to formal NLI systems whose multi-stage pipelines produce compounding ambiguity. As such systems adopt richer cross-stage linguistic representations, the exploration paradigm is likely to become increasingly relevant.

More broadly, formal NLI systems such as *lightblue* represent a bridge between formal linguistics and computational approaches to language understanding: they ground inference in well-defined linguistic theories, making theoretical assumptions explicit and computationally testable. Such systems also hold promise as platforms for evaluating the reasoning capabilities of neural language models against rigorous formal linguistic standards. However, if the development workflow for the systems that embody this bridge remains impractical, the bridge itself cannot be crossed. By making the development workflow of formal NLI tractable and transparent, *Express* helps ensure that the connection between formal linguistics and NLP is not merely theoretical but practically sustainable.

Acknowledgments

This work was supported in part by JST CREST Grant Number JPMJCR2565 and JSPS KAKENHI Grant Number JP23H03452, Japan.

References

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.

Daisuke Bekki and Koji Mineshima. 2017. Context-passing and underspecification in dependent type semantics. In *Modern Perspectives in Type-Theoretical Semantics*, Studies in Linguistics and Philosophy, pages 11–41. Springer.

Johan Bos. 2008. Wide-coverage semantic analysis with Boxer. In *Semantics in Text Processing. STEP 2008 Conference Proceedings*, pages 277–286. College Publications.

Stergios Chatzikyriakidis and Zhaohui Luo. 2014. Natural language inference in Coq. *Journal of Logic, Language and Information*, 23(4):441–480.

Dick Crouch, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, John Maxwell, and Paula Newman. 2011. XLE documentation. Palo Alto Research Center. Available at https://ling.sprachwiss.uni-konstanz.de/pages/xle/doc/xle_toc.html.

Hinari Daido and Daisuke Bekki. 2020. Development of an automated theorem prover for the fragment of DTS. In *Proceedings of the 17th International Workshop on Logic and Engineering of Natural Language Semantics (LENLS17)*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Hai Hu, Qi Chen, and Larry Moss. 2019. Natural language inference with monotonicity. In *Proceedings of the 13th International Conference on Computational Semantics - Short Papers*, pages 8–15, Gothenburg, Sweden. Association for Computational Linguistics.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico. Association for Computational Linguistics.

Per Martin-Löf. 1984. *Intuitionistic Type Theory*, volume 9. Bibliopolis, Naples.

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.

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.

Michael Snoyman. 2015. *Developing Web Apps with Haskell and Yesod*, 2nd edition. O'Reilly Media.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press.

Asa Tomita, Mai Matsubara, Hinari Daido, and Daisuke Bekki. 2025. Natural language inference with CCG parser and automated theorem prover for DTS. In *Proceedings of the Second Workshop on the Bridges and Gaps between Formal and Computational Linguistics (BriGap-2)*, pages 1–7, Düsseldorf, Germany. Association for Computational Linguistics.

Asa Tomita, Hitomi Yanaka, and Daisuke Bekki. 2024. Reforging: A method for constructing a linguistically valid Japanese CCG treebank. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 196–207, St. Julian's, Malta. Association for Computational Linguistics.

A Appendix - Supplementary Details

A.1 The Branching Problem in Other Formal NLI Systems

ccg2lambda (Martínez-Gómez et al., 2016) translates CCG derivations into first-order or higher-order logical formulas and delegates inference to an off-the-shelf theorem prover. Because its semantic representations do not involve underspecified terms requiring anaphora resolution during type checking, the exponential branching across sentences that characterizes *lightblue*'s DTS-based pipeline does not arise. LangPro (Abzianidze, 2017) similarly resolves lexical and phrasal ambiguities within a natural logic tableau framework where branching is managed by the prover's own search strategy rather than by an external multi-stage pipeline. The branching problem addressed in this paper is thus particularly acute in systems where type checking and anaphora resolution are interleaved across sentences, as in DTS-based architectures.

A.2 User Study Details

Participant expertise. All four participants are researchers at the authors' institution with experience in grammar development using *lightblue*. Table 4 summarizes their experience and research focus.

ID	Exp.	Research focus
P1	3 yrs	NLI system evaluation using formal test suites
P2	1 yr	Neural-symbolic integration for formal NLI
P3	3 yrs	CCG-based semantic composition for Japanese
P4	2 yrs	Neural acceleration of theorem proving

Table 4: Participant expertise. Exp.: years of experience with *lightblue*.

Likert scale questionnaire. Participants rated each item on a 5-point scale (1: strongly disagree, 5: strongly agree). Q1 and Q2 were administered for both conditions; Q3 and Q4 were administered for *Express* only, as they explicitly compare *Express* against the baseline.

Q1 It was easy to reach the target inference path.

Q2 It was easy to understand the inference process.

Q3 *Express* reduced the workload compared to the conventional approach.

Q4 I would like to use *Express* in actual grammar development work.

A.3 NLI Problems Used in the Evaluation

Table 5 shows the two NLI problems used in §5.1 and §5.2. Both problems consist of one premise and one hypothesis (k=2) and involve entailment relations concerning Japanese passive constructions, where branching arises from the syntactic analysis and anaphora resolution of passive forms.

	Problem A	**Problem B**
Premise	*Taro-ga Hanako-ni nak-are-ta.* (Taro was cried on by Hanako.)	*Taro-ga Hanako-o shikat-ta.* (Taro scolded Hanako.)
Hypothesis	*Hanako-ga nai-ta.* (Hanako cried.)	*Hanako-ga shikar-are-ta.* (Hanako was scolded.)
Label	Yes	Yes

Table 5: NLI problems used in the evaluation (§5).

Problem A involves an indirect passive (*adversative passive*) construction, while Problem B involves entailment from an active to a passive sentence. Under exhaustive execution, Problem A generated 260 disambiguation paths and Problem B generated 128 (Table 1).

Figure 6 shows a portion of the HTML output generated by *lightblue*'s conventional exhaustive execution for Problem A, as used in the baseline condition of the user study. The small scroll bars indicate that the visible area represents only a fraction of the full output.

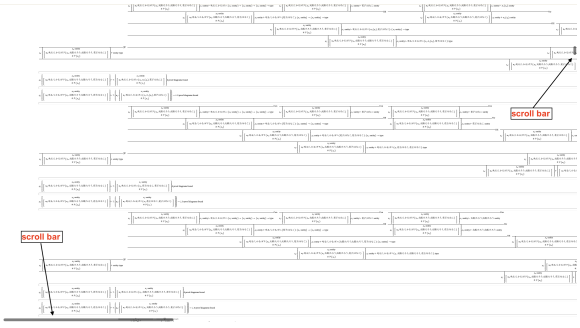


Figure 6: Conventional *lightblue* output for Problem A (3.7 MB HTML file). The developer must manually locate a single path of interest within this voluminous output. Compare with the *Express* interface in Figures 3 and 4.

Neural Wani: Toward Accelerating the Automated Theorem Prover *wani* for Dependent Type Theory

Nanako Miyagawa and Hinari Daido* **and Daisuke Bekki**
Ochanomizu University
{miyagawa.nanako, daido.hinari, bekki}@is.ocha.ac.jp

Abstract

This paper proposes *Neural Wani*, an integration of a neural model into the automated theorem prover *wani* for Dependent Type Theory (DTT), aimed at accelerating proof search in natural language inference (NLI) pipelines. We implemented a lightweight LSTM-based model to predict the probability distribution of applicable inference rules and integrated it into *wani*'s backward inference process. Evaluation using the JSeM dataset demonstrates that *Neural Wani* achieves a 1.41× speedup compared to the standard non-neural baseline. Although slight overhead is observed in simpler proofs, our results indicate that neuro-symbolic integration effectively guides search in complex DTT-based automated theorem proving.

1 Introduction

Elucidating the computational processes underlying the structural meaning of natural language is a foundational goal in computational linguistics. While recent data-driven methods have shown remarkable performance, pipeline systems based on formal linguistics remain crucial for achieving strict interpretability and verifiable reasoning in natural language inference (NLI). Throughout this paper, NLI denotes the task of deciding, for a set of premises and a candidate conclusion, whether the premises entail the conclusion, contradict it, or do neither. Because clarifying these linguistic processes through manual derivation is prohibitively complex, establishing robust, automated NLI pipelines is in high demand.

To this end, Tomita et al. (2025) developed a comprehensive NLI pipeline integrating *lightblue* (Bekki and Kawazoe, 2016)[1]—a CCG-based syntactic and semantic parser—with *wani* (Daido and Bekki, 2020), an automated theorem prover designed for Dependent Type Theory (DTT) (Martin-Löf, 1984). This system processes raw text through syntactic and semantic analysis, type checking, anaphora and presupposition resolution, and automated theorem proving.

A key strength of this architecture is its highly transparent syntax-semantics interface, afforded by CCG (Steedman, 1996), which compositionally maps surface strings into complex DTS semantic representations. Despite this structural elegance, the practical deployment of such a pipeline is severely constrained by the computational cost of the final phase: automated theorem proving. The vast search space inherent to DTT makes purely logic-based proof search inefficient, necessitating a more guided approach.

1.1 *wani*

Wani, the automated theorem prover integrated into the aforementioned NLI system, adopts DTT as its proof system. Concretely, *wani* is based on a natural-deduction-style formulation of DTT, in which each type former is governed by formation (F), introduction (I), and elimination (E) rules, and hypotheses may be introduced and later discharged within a derivation. This follows the standard natural-deduction presentation of Martin-Löf's intuitionistic type theory (Martin-Löf, 1984), on which DTT is based. A judgment in DTT is expressed in the following form:

$$\Gamma \vdash M : A$$

where Γ represents the context, M the term, and A the type. From the perspective of type theory, this judgment signifies that "the term M has type A under the context Γ." Simultaneously, from the viewpoint of proof theory, it indicates that "there exists a proof M for the conclusion A given the premises Γ" via the Curry-Howard correspondence.

*This work was conducted independently and does not reflect the views or positions of Amazon.

[1]https://github.com/DaisukeBekki/lightblue/

Third Workshop on the Bridges and Gaps between Formal and Computational Linguistics, pages 12–21

A judgment in which the proof term M is unknown, as shown below, is called a *proof search query*, and the task of finding a proof term for such a query is referred to as *proof search*.

$$\Gamma \vdash ? : A$$

To resolve a proof search query, *wani* employs a combination of forward and backward inference. Forward inference focuses on "what can be derived from the premises," constructing a proof tree from the top down. Conversely, backward inference focuses on "what is required to prove a given proposition," constructing the proof tree from the bottom up. In this process, inference rules are applied to the target proposition to identify the necessary conditions as subgoals.

1.2 Dependent Type Semantics (DTS)

The natural language inference system described in Section 1 adopts Dependent Type Semantics (DTS) (Bekki, 2014; Bekki and Mineshima, 2017) as its framework for semantic representation and composition. DTS is a theory of natural language semantics based on DTT. In DTT, Π-types and Σ-types are used as generalizations of function types and product types, denoted as $(x : A) \to B$ and $(x : A) \times B$, respectively. Since the type of the consequent B can depend on the proof of the antecedent A, DTS is capable of representing sentence meanings that depend on the meanings of preceding sentences. This property allows DTS to handle complex linguistic phenomena such as anaphora resolution and presupposition binding.

Furthermore, DTS is a form of proof-theoretic semantics, which defines the meaning of a sentence as its verification condition mediated by inference rules. Compared to model-theoretic semantics—the standard theory in formal semantics—DTS offers the implementation advantage of computing the success or failure of inference solely through proof search, without the need to refer to external models. However, DTT—the higher-order foundation of DTS—possesses greater expressive power than first-order logic but faces computational challenges such as undecidability. To address this, we build upon the framework of *Neural Wani* (Miyagawa et al., 2025), introducing a neuro-symbolic integration to reduce *wani*'s computation time by optimizing proof search efficiency.

2 Background

Neural Wani is an initiative aimed at accelerating inference by integrating neural models into the proof search process of DTT. The traditional backward inference in *wani* relies on depth-first search (DFS) with predefined depth and time limits. This approach faces a significant challenge: computation time increases exponentially if inappropriate inference rules are applied during the early stages of the search. To address this, *Neural Wani* employs a method that takes a DTT judgment as input and uses a neural model to predict the most applicable inference rule. By prioritizing the search paths with higher predicted probabilities, the system substantially reduces the overall time required for proof search.

In a preliminary study (Miyagawa et al., 2025), various vector embedding methods for DTT judgments were compared and evaluated. Based on those evaluation results, an embedding strategy specifically optimized for the structural requirements of DTT proof search was established, serving as the foundational representation for the predictive model in this work.

Although *Neural Wani* originates from this preliminary study (Miyagawa et al., 2025), the present work extends it in three respects. **(i) Task setting:** the prior model predicted inference rules from judgments that still contained proof terms, whereas we address the strictly harder and practically essential setting of predicting rules directly from *proof search queries*—judgments that lack proof terms (Section 3.1). **(ii) Integration:** rather than evaluating the predictor in isolation, we embed it into *wani*'s actual backward inference loop, so that the predicted rule ordering directly governs the proof search. **(iii) Evaluation:** we measure the resulting end-to-end speedup on an NLI pipeline using held-out JSeM problems, instead of reporting classification accuracy alone (Section 4.3).

3 Proposed Method

3.1 Predicting Rules from Proof Search Queries

Previous work (Miyagawa et al., 2025) developed a model to predict inference rules using judgments that already contained proof terms as input. In DTT, a proof term explicitly encodes the application history of inference rules; consequently, predicting rules from such judgments is relatively straightforward, as evidenced by a reported high micro-F1

score of 0.982. However, during the actual process of proof search (backward inference), as discussed in Section 1.1, the proof term is unknown. A practical automated prover must therefore be capable of predicting inference rules directly from proof search queries, which are judgments lacking proof terms.

In backward inference, applying an inference rule to a current judgment generates new subgoals, which are themselves judgments without proof terms. By recursively predicting the appropriate rule names at each of these stages, the entire proof tree can be constructed. Therefore, the task addressed in this study—predicting rules under the condition where the proof term is missing—marks a crucial step toward achieving effective automated search in *Neural Wani*.

To make the setting concrete, consider a toy NLI problem. We use a Japanese example, in keeping with our JSeM-based evaluation, transliterated in romaji with English glosses in parentheses: from the premises "Subete no gakusei ga aruita" ("Every student walked") and "John wa gakusei de aru" ("John is a student"), the hypothesis "John wa aruita" ("John walked") follows. For readability we use a simplified DTS representation in which $\mathbf{student}$ and $\mathbf{walk}$ are treated as unary predicates over entities; the representations that *lightblue* actually produces additionally encode event variables, tense, and multi-place predicate–argument structure, which we abstract away here.[2] Over a signature declaring $\mathbf{student}, \mathbf{walk} : \mathsf{entity} \to \mathsf{type}$ and $\mathbf{john} : \mathsf{entity}$, the premises correspond to a context Γ, and the prover attempts the query $\Gamma \vdash ? : \mathbf{walk}(\mathbf{john})$. Figure 1 shows the resulting proof, which *wani* builds by backward inference using only (ΠE) and $(\mathrm{Membership})$; since a proof term is found, the verdict is YES (entailment).

3.2 Tokenization and De-lexicalization

To handle the open vocabulary of natural language and focus on the underlying logical structure of DTT judgments, we implemented a de-lexicalization strategy. Instead of relying on raw surface strings or high-dimensional word embeddings, we map each element of a judgment to a predefined set of abstract tokens:

- **Structural Tokens**: Logical constants (e.g., Π, Σ, $=$, `type`) and delimiters (e.g., `EOCon`, `EOSig`) are explicitly assigned unique tokens to preserve the formal syntax of DTT.

- **Constant Placeholders**: Lexical constants appearing in a judgment are sequentially mapped to generic tokens `Word1` through `Word31` based on their order of appearance. This threshold was empirically determined to cover the majority of our dataset; any constants exceeding this limit are mapped to an `UNKNOWN` token.

- **Variable Placeholders**: Bound variables are similarly mapped to `Var'0` through `Var'6` based on De Bruijn indices, with any overflow mapped to `Var'unknown`. This mapping naturally resolves α-equivalence issues, allowing the model to treat structurally identical terms with different variable names as the same sequence.

This approach ensures that the model remains agnostic to specific lexical semantics, compelling it to learn the abstract structural templates that drive proof search. Because lexical content is abstracted away in this manner, the predictor depends solely on the structural form of DTT judgments. It is therefore independent of the surface language analyzed upstream by *lightblue*, and, although our experiments target NLI, the same mechanism applies to any proof search query expressible in DTT.

3.3 Embedding Representation of Judgments

In this study, we employ the "EO~delimiter" method to encode proof search queries for the neural model. In a previous evaluation of embedding strategies (Miyagawa et al., 2025), comparing four embedding methods—parentheses delimiters, SEP delimiters, EO~delimiters, and no delimiters—the EO~delimiter method achieved the highest micro-F1 score. Furthermore, because the current task is strictly more complex due to the absence of proof terms, explicitly demarcating the logical roles and boundaries of each element within the judgment is essential. Therefore, the EO~delimiter representation is particularly well suited to this setting.

The specific data structure is as follows. Because the proof term is inherently unknown during the proof search, the `term` component is excluded from the input representation. For the goal judgment

[2] For instance, *lightblue* renders "John walked" as $(x{:}\mathsf{entity}) \times (\mathbf{walk}(x, \mathbf{john}, @\mathsf{entity})) \times (\mathbf{Past}(x))$ rather than the simplified $\mathbf{walk}(\mathbf{john})$ used here.

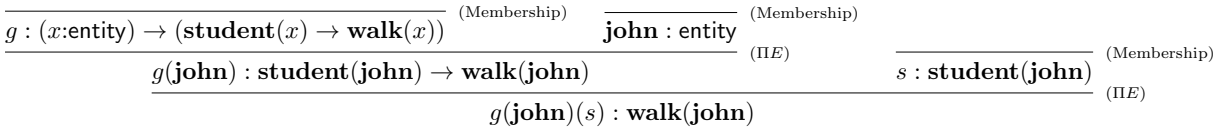


Figure 1: A toy NLI proof built by *wani*: from "Subete no gakusei ga aruita" ("Every student walked") and "John wa gakusei de aru" ("John is a student"), it derives "John wa aruita" ("John walked"; $\mathbf{walk}(\mathbf{john})$) by backward inference. The representation is simplified for illustration (see the main text). Leaves labeled (Membership) retrieve premises from the context and constants from the signature.

$\Gamma \vdash ? : \mathbf{walk}(\mathbf{john})$ of the toy proof in Figure 1, the components are:

```
signature = [(student, entity->type),
(walk,     entity->type),     (john,
entity)]

context = [student(john),
(x:entity)->(student(x)->walk(x))]

~~term = ...~~    ← Excluded in this study

type = walk(john)
```

After de-lexicalization, which replaces the constants `student`, `walk`, and `john` with `Word1`, `Word2`, and `Word3` (and the bound variable x with `Var'0`/`Var'1` according to its De Bruijn index), this judgment is encoded as the following EO~delimiter token sequence (grouped by component):

```
Word1 COMMA Pi' Entity' EOPre Type' EOPre
EOPre EOPair
Word2 COMMA Pi' Entity' EOPre Type' EOPre
EOPre EOPair
Word3 COMMA Entity' EOPre EOPair EOSig
App' Word1 EOPre Word3 EOPre EOPre
Pi' Entity' EOPre Pi' App' Word1 EOPre
Var'0 EOPre EOPre App' Word2 EOPre Var'1
EOPre EOPre EOPre EOPre EOCon
App' Word2 EOPre Word3 EOPre EOPre EOTyp
```

3.4 Neural Model Architecture

Following previous work (Miyagawa et al., 2025), we constructed the rule prediction model using Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). The specific hyperparameters used in our experiments are detailed in Table 1.

Table 1: Hyperparameters

Number of Epochs	10
Learning Rate	5×10^{-4}
Number of Layers	1
Dropout	None
Input Size	128
Hidden State Size	128
BPTT Steps	32

While Transformer-based architectures (Vaswani et al., 2017) have become the standard in various sequence-to-sequence tasks, we deliberately opted for a lightweight LSTM for our rule prediction task. The primary rationale lies in the strict latency constraints of automated theorem proving. In backward inference, the model is queried recursively at every node of the expansive search tree. The heavy multi-head attention mechanisms of Transformers introduce significant computational overhead, which could easily negate the speedup gained from pruning the search space.

Furthermore, unlike general natural language processing, where long-range global context or external discourse information may alter the interpretation, the "context" required for rule prediction is entirely self-contained within the localized structure of the judgment itself. Given this structural locality, the sequential processing capability of a lightweight LSTM suffices, balancing structural pattern recognition against the low-latency inference required to accelerate the prover.

4 Evaluation Experiments

4.1 Training and Evaluation Dataset

We used Natural Language Inference (NLI) problems collected from JSeM (Kawazoe et al., 2015a,b), a Japanese semantic test suite, as our dataset. For each problem, we performed CCG-based syntactic parsing using the *lightblue* parser, followed by semantic composition within the DTS framework. The resulting type-checked proof diagrams serve as the primary data for this study. We selected eight specific sections to ensure coverage across a broad range of linguistic phenomena, including generalized quantifiers, verbs, adjectives, compound adjectives, propositional attitudes, nominal anaphora, noun phrases, and temporal reference.

In this experiment, to prevent data leakage across identical proof contexts, we allocated 90% of the

676 valid JSeM problems for model training, validation, and testing, strictly reserving the remaining 10% for a downstream comparative performance analysis between *Neural Wani* and the baseline *wani* system. From the allocated 90% problem set, we extracted the training data according to the following procedure:

1. We performed type checking with *wani* to generate proof trees.
2. We extracted pairs consisting of a judgment and its corresponding applied inference rule name from the resulting proof trees.
3. From the extracted pairs, we selected only the rules used in *wani*'s backward inference, explicitly excluding pairs corresponding to formation rules.
4. To construct the final dataset, we collected up to 2,000 data points per selected rule. For rules with fewer than 2,000 available instances, we used all available data.

The capping in Step 4 is crucial for addressing the severe class imbalance inherent in proof search; majority rules such as $(\mathrm{Membership})$[3] and (ΠE) occur significantly more often than others. By limiting these dominant classes, we mitigated model bias and ensured more robust generalization across the rules that occur in the data; Figure 2 contrasts the distribution before and after capping. Although Table 2 lists ten backward-inference rules, the training data are drawn only from non-formation rules (Step 3 above excludes formation rules), and among these, only (ΠI), (ΠE), $(\mathrm{Membership})$, and (ΣI) actually occur in the proof trees extracted from our eight JSeM sections. Rules such as $(\top I)$, (DNE), and (EFQ) do not appear in this data and are therefore absent from both the plot and the per-label results in Table 4. Among the four observed rules, (ΣI) is extremely rare (only two instances), so in practice the model learns to distinguish the three remaining rules.

Through this controlled procedure, we obtained a total of $4,689$ judgment–rule pairs. We partitioned this final dataset into training (80%), validation (10%), and test (10%) sets, using a stratified split so that the per-rule label proportions are identical across all three subsets.

[3] $(\mathrm{Membership})$ is not a primitive rule of DTT but an implementation-level rule of *wani*: it proves a judgment by looking up a matching declaration, covering two cases—retrieving a constant from the signature, which corresponds to DTT's (CON) rule, and retrieving an assumption from the context, which is not given a named rule in standard DTT formulations.

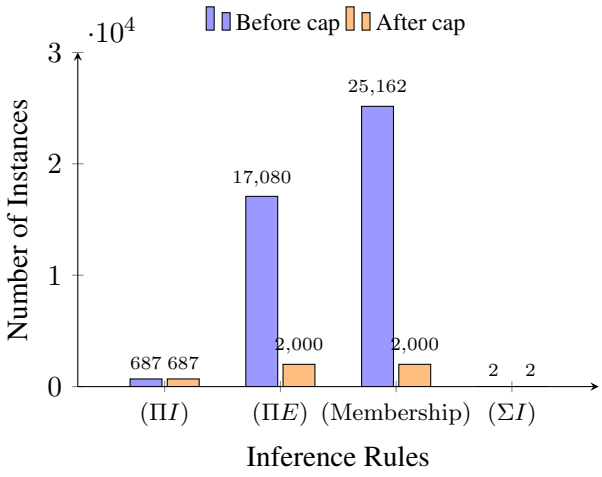


Figure 2: Label distribution over the four backward-inference rules that occur in the dataset, shown before and after the 2,000-instance cap. Only (ΠE) and $(\mathrm{Membership})$ exceed the cap, whereas (ΠI) and (ΣI) remain unchanged.

4.1.1 Rationale for Focusing on Backward Inference Rules

The primary objective of *Neural Wani* is to use neural models to improve the efficiency of computationally expensive stages within the proof search process. In *wani*'s inference system, the rules applicable to forward reasoning are strictly limited, whereas the candidate rules applicable at each step of backward inference are far more diverse, as shown in Table 2. Furthermore, certain rules—such as (ΠE) and (DNE)—can be applied to any type without restriction, which frequently leads to an explosive expansion of the search space. Because these backward inference steps inherently involve a vast search space and represent the primary computational bottleneck, this study focuses specifically on predicting the inference rules used during backward inference.

Table 2: Inference rules in *wani*, grouped by direction. $(\mathrm{Membership})$ appears in both the forward and backward sets.

Forward	**Backward**		
$(\mathrm{Membership})$	(ΠI)	(ΣI)	$(\mathrm{Membership})$
(ΣE)	(ΠE)	(ΣF)	(DNE)
$(=I)$	(ΠF)	$(\top I)$	(EFQ)
$(=E)$	$(=F)$		

4.1.2 Rationale for Excluding Formation Rules

Formation rules in DTT are represented as follows:

$$\frac{\begin{array}{c}\overline{x : A}^{\,i} \\ \vdots \\ A : \mathsf{s}_1 \qquad B : \mathsf{s}_2\end{array}}{(x : A) \to B : \mathsf{s}_2}(\Pi F), i$$

$$\text{where } (\mathsf{s}_1, \mathsf{s}_2) \in \{(\mathsf{type}, \mathsf{type}), (\mathsf{type}, \mathsf{kind}), (\mathsf{kind}, \mathsf{kind})\}$$

Figure 3: Example of the (ΠF) rule

Following the natural-deduction convention, the overlined assumption $\overline{x : A}^{\,i}$ in Figure 3 denotes a hypothesis that is discharged at the inference step labeled i.

$$\frac{A : \mathsf{type} \quad M : A \quad N : A}{M =_A N : \mathsf{type}}(=F)$$

Figure 4: Example of the $(=F)$ rule

In DTT, the conclusion (the lower part) of a formation rule typically yields a judgment with either type or kind as its type. When a judgment possesses these specific types, the applicable rule is syntactically and uniquely determined, with few exceptions such as (CON). In other words, the application of formation rules is deterministic and does not necessitate the probabilistic prediction provided by a neural model. For this reason, we explicitly excluded formation rules from the scope of our prediction task and evaluation experiments.

4.2 Evaluation Methodology for *Neural Wani*

We integrated the rule prediction model into the baseline *wani* system to investigate whether it effectively improves the search.[4] For this evaluation, we randomly selected 50 NLI proof search problems from the reserved data strictly excluded from the training, validation, and testing sets.

The evaluation process for *wani* was conducted according to the format illustrated in Figure 5. Specifically, *wani* returns YES—meaning that the premises entail the conclusion—if a proof term for the problem $\Gamma \vdash ? : A$ is discovered, and NO—meaning that the premises contradict the conclusion—if a proof term for the negation $\Gamma \vdash ? : \neg A$ is found instead. If the system fails to find either within the specified time limit,[5] it returns UNK (Unknown), the case in which neither entailment nor contradiction can be established. For this evaluation, *wani* was configured to operate in intuitionistic mode with a search depth limit of 9. *wani* provides two mutually exclusive modes—an intuitionistic mode that uses (EFQ) and a classical mode that uses (DNE)—so the two rules are never available simultaneously; accordingly, (DNE) is not used in our experiments.

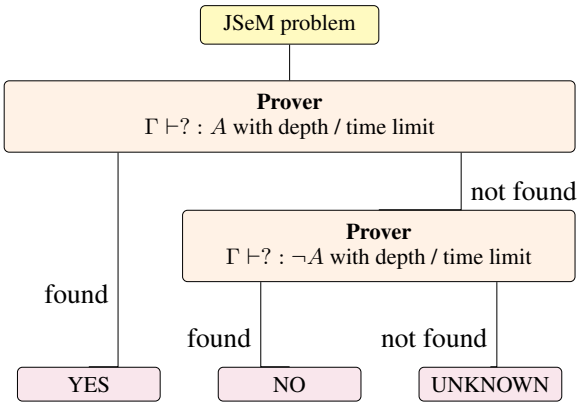


Figure 5: Evaluation Method

Experimental precautions. To ensure a fair comparison and minimize experimental bias, several technical precautions were implemented. First, to account for lazy evaluation in Haskell (the implementation language of *wani*), we ensured that proof search queries were fully evaluated and manual garbage collection was triggered before each test case to prevent resource interference between runs. Second, to mitigate potential system-level execution biases, we alternated the execution order: for half of the test problems, the baseline *wani* was executed first, while for the other half, *Neural Wani* was executed first. Finally, while *Neural Wani* employs a caching mechanism to store rule predictions for recurring judgments within a single proof search, this cache was strictly cleared between experiments under different time limits to ensure the independence of each measurement.

[4] During the search process, formation rules (which were excluded from the training data) are applied first. Subsequently, *Neural Wani* predicts the priority of backward inference rules, and the search proceeds according to this predicted order. To fully assess the model's predictive capability, we did not perform top-k pruning; rather, all applicable rules were considered in the sorted order.

[5] In *wani*, we impose depth and time limits on the deduce function, which performs a single backward inference.

4.3 Experimental Results

4.3.1 Model Evaluation

Table 3: Overall classification performance

	Prec	Rec	F1	Supp
micro	0.821	0.821	0.821	470
macro	0.834	0.608	0.591	470
w-avg	0.834	0.821	0.821	470

Table 4: Classification performance per label

	Prec	Rec	F1	Supp
(ΠI)	0.582	0.768	0.662	69
(ΠE)	0.866	0.710	0.780	200
(Membership)	0.888	0.955	0.920	200
(ΣI)	1.000	0.000	0.000	1

4.3.2 Performance Comparison between Normal and Neural Approaches

Table 5: Performance comparison between the baseline *wani* (Normal) and *Neural Wani* (Neural) across time limits. Out of 50 NLI problems, both systems solved 8 under every limit, so the success rate is identical and we report only timing. "all" averages over all 50 problems (including timeouts), whereas "both succ." averages only over problems solved by both systems. Speedup is the ratio of Normal to Neural time.

Time limit	Avg. time	Normal	Neural	Speedup
30s	all (sec)	180.619	128.376	1.41×
	both succ. (sec)	2.093	4.915	0.43×
60s	all (sec)	309.516	219.143	1.41×
	both succ. (sec)	2.068	2.087	0.99×
90s	all (sec)	404.035	314.057	1.29×
	both succ. (sec)	2.062	2.082	0.99×

5 Discussion

5.1 Neural Model

As shown in Table 3, the overall evaluation yielded a high micro-F1 score of 0.821, whereas the macro-F1 score remained at 0.591. This discrepancy stems from the imbalanced label distribution within the proof search dataset. Since the micro-F1 score reflects the overall accuracy across all instances, it is heavily influenced by the model's performance on majority classes. In contrast, the lower macro-F1 score—which treats all classes equally—indicates that predicting minority rules remains a challenge. We further note that this micro-F1 of 0.821 is lower than the 0.982 reported by the prior model that operated on judgments containing proof terms (Miyagawa et al., 2025). This gap is expected rather than a regression: our setting removes the proof term—the single most informative cue for identifying the applied rule—and therefore constitutes a substantially harder prediction problem, as discussed in Section 3.1.

However, from the perspective of our primary objective—accelerating the proof search pipeline with *Neural Wani*—this performance profile is highly advantageous for the following reason.

The robust micro-F1 score of 0.821 directly translates to overall search efficiency. In DTT proof search, the majority of judgments involve high-frequency rules such as (Membership) and (ΠE). Maintaining high predictive accuracy for these dominant rules ensures that, at most nodes in the search tree, the model correctly prioritizes the optimal inference rule. This successful pruning directly reduces the exponentially growing computation time of depth-first search.

In conclusion, while there is room for improvement in predicting long-tail rules, the proposed model demonstrates high practical performance as a supportive tool for accelerating the DTT prover *wani*.

5.2 Quantitative Analysis of Search Efficiency

As shown in Table 5, *Neural Wani* maintains exactly the same success rate as the non-neural baseline *wani*: both systems solved 8 out of 50 highly complex NLI problems under all time limits. This result confirms that the integration of the neural model safely preserves the logical completeness of the original prover; problems solvable by the baseline remain solvable with *Neural Wani*.

While the absolute success rate remains unchanged, *Neural Wani* demonstrates a consistent reduction in average execution time across all test cases, including those that ultimately resulted in timeouts. Notably, the overall average runtime achieved a speedup of up to 1.41× under the 30s and 60s limits. This marked acceleration suggests that the neural-guided prioritization effectively steers the prover away from exploring deep, unproductive branches in the search space.

Conversely, an analysis of problems that are solved rapidly by both systems reveals a well-known neuro-symbolic trade-off. In structurally shallow cases, *Neural Wani* occasionally incurs

slight overhead due to the constant inference time of the LSTM. This indicates that when the search space is shallow, the computational cost of neural evaluation may not be entirely offset by the benefit of heuristic pruning.

Taken together, these observations confirm that *Neural Wani* is well suited to its design goal: it substantially improves search efficiency in large, intractable search spaces where traditional provers struggle, while introducing only a modest overhead in already trivial cases.

5.3 Cognitive Parallelism: Intuition-Guided Search

Beyond the quantitative gains, the integration of a neural model into *wani*'s symbolic engine admits an interpretive analogy—we stress that it is an analogy, not a cognitive claim—through Dual Process Theory (Evans and Frankish, 2009): the LSTM classifier functions as **System 1**, providing rapid heuristic suggestions from the structural features of a judgment, while *wani* acts as **System 2**, verifying them through backtracking and type checking. What makes this synergy specific to *Neural Wani*—rather than generic to any neuro-symbolic prover—is the locus at which System 1 intervenes, namely the order in which inference rules are expanded during backward search; we contrast this locus with premise selection and differentiable proving in Section 6. These results are consistent with this interpretation: the predictor attains its highest accuracy on $(\mathrm{Membership})$ and (ΠE) (Table 4), and *Neural Wani* achieves a $1.41\times$ average speedup in the regime where the baseline tends to time out (Table 5).

6 Related Work

Logic-based NLI and automated reasoning. Combining compositional semantics with automated reasoning to perform NLI has a long tradition. Blackburn and Bos (2005) established the canonical pipeline of computational semantics, in which surface strings are compositionally mapped to first-order logic via λ-calculus, and semantic judgments such as entailment and consistency are decided by running off-the-shelf first-order theorem provers and model builders in parallel; *wani*'s YES/NO/UNK procedure (Figure 5) follows the same entailment/contradiction/unknown decision structure. This model-theoretic line was scaled to real-world text through wide-coverage CCG-based semantic construction (Bos et al., 2004). A complementary, proof-theoretic tradition instead establishes entailment by symbolic inference rather than by checking models: natural-logic-based systems (MacCartney and Manning, 2008), tableau-based systems (Abzianidze, 2015), and Coq-based systems (Chatzikyriakidis and Luo, 2014; Mineshima et al., 2015; Martínez-Gómez et al., 2017; Chatzikyriakidis and Bernardy, 2019). Our work belongs to this latter, proof-theoretic branch: in contrast to the model-theoretic tradition that verifies inferences against external models, DTS computes entailment purely through proof search (Section 1.2), and we accelerate the dedicated DTT prover *wani* by guiding its proof search with a neural model rather than leaving it to search purely symbolically.

Neural guidance for theorem proving. Prior attempts to bring learning into theorem proving are largely distinguished by which component of the prover the learned model informs. A first and well-studied locus is premise selection, where a model scores how likely each available fact is to contribute to a proof: Wang et al. (2017) learn deep graph embeddings of formulas for this purpose, and Kogkalidis et al. (2024) pursue the same goal for dependently typed Agda programs via structure-aware encoders. A second locus is the inference engine itself, which can be made differentiable so that symbol-level similarities are learned end to end, as in the differentiable prover of Rocktäschel and Riedel (2017) for knowledge-base completion. *Neural Wani* departs from both: the premises of an NLI problem are already supplied explicitly by the upstream semantic-composition stage, so premise selection is not the bottleneck, and *Neural Wani* leaves *wani*'s symbolic inference engine unchanged rather than replacing it with a differentiable approximation. Instead, our learned model operates at a third locus—the order in which inference rules are expanded during backward search—thereby steering the search while leaving the symbolic core of the prover, and hence its soundness, intact.

7 Conclusion

In this study, we developed *Neural Wani*, which optimizes the proof search process of the DTT-based theorem prover *wani* through neural rule prediction. By focusing on judgments lacking proof terms, we addressed a critical bottleneck in practi-

cal automated theorem proving. Our experimental results confirmed that prioritizing inference rules based on predicted probabilities significantly reduces search time in a natural language inference pipeline, achieving a maximum average speedup of 1.41×.

Limitations

A primary limitation of this study is the discrepancy between the training data and the actual task environment. Ideally, a model for accelerating proof search should be trained on successful proof search trajectories. However, due to the current scarcity of solved proof search instances in complex DTT-based NLI, we used existing type-checking data as a proxy for training. While this allows the model to learn the structural relationships between judgments and rules, there is an inherent distribution shift between type checking and backward inference.

To address this, several concurrent projects (Tomita and Bekki, 2026; Saeki et al., 2026; Sakuma et al., 2026; Kobayashi et al., 2026; Matsubara et al., 2026) are currently underway to expand the coverage of solvable proof search problems within the *wani* ecosystem. We expect that as the volume of successfully solved cases increases, we will be able to refine the model using authentic proof search data, further narrowing the gap between training and real-world application.

Acknowledgments

This work was supported in part by JST CREST Grant Number JPMJCR2565 and JSPS KAKENHI Grant Number JP23H03452.

References

Lasha Abzianidze. 2015. Towards a wide-coverage tableau method for natural logic. In New Frontiers in Artificial Intelligence: JSAI-isAI 2014 Workshops, LENLS, JURISIN, and GABA, Revised Selected Papers, volume 9067 of Lecture Notes in Artificial Intelligence, pages 66–82. Springer.

Daisuke Bekki. 2014. Representing anaphora with dependent types. In Logical Aspects of Computational Linguistics, volume 8535 of Lecture Notes in Computer Science, pages 14–29. Springer.

Daisuke Bekki and Ai Kawazoe. 2016. Implementing variable vectors in a CCG parser. In Logical Aspects of Computational Linguistics, pages 52–67. Springer.

Daisuke Bekki and Koji Mineshima. 2017. Context-passing and underspecification in dependent type semantics. In Stergios Chatzikyriakidis and Zhaohui Luo, editors, Modern Perspectives in Type-Theoretical Semantics, pages 11–41. Springer.

Patrick Blackburn and Johan Bos. 2005. Representation and Inference for Natural Language: A First Course in Computational Semantics. CSLI Studies in Computational Linguistics. CSLI Publications, Stanford, CA.

Johan Bos, Stephen Clark, Mark Steedman, James R. Curran, and Julia Hockenmaier. 2004. Wide-coverage semantic representations from a CCG parser. In COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics, pages 1240–1246, Geneva, Switzerland. COLING.

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.

Stergios Chatzikyriakidis and Zhaohui Luo. 2014. Natural language inference in Coq. Journal of Logic, Language and Information, 23:441–480.

Hinari Daido and Daisuke Bekki. 2020. Development of an automated theorem prover for the fragment of DTS. In the 17th International Workshop on Logic and Engineering of Natural Language Semantics.

Jonathan Evans and Keith Frankish. 2009. In two minds: Dual processes and beyond. Oxford University Press.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural Computation, 9(8):1735–1780.

Ai Kawazoe, Ribeka Tanaka, Koji Mineshima, and Daisuke Bekki. 2015a. A framework for constructing multilingual inference problem sets: Highlighting similarities and differences in semantic phenomena between English and Japanese. In 1st International Workshop on the Use of Multilingual Language Resources in Knowledge Representation Systems.

Ai Kawazoe, Ribeka Tanaka, Koji Mineshima, and Daisuke Bekki. 2015b. An inference problem set for evaluating semantic theories and semantic processing systems for Japanese. In Proceedings of the Twelfth International Workshop on Logic and Engineering of Natural Language Semantics, pages 67–73.

Honoka Kobayashi, Hinari Daido, and Daisuke Bekki. 2026. Neural DTS: Shizen-gengo-suiron-shisutemu-e-no sōkyoku-bunruiki-no kumikomi (trans. Neural DTS: Incorporating hyperbolic classifiers into natural language inference systems). In Proceedings of the 32nd Annual Conference of the Association for Natural Language Processing.

Konstantinos Kogkalidis, Orestis Melkonian, and Jean-Philippe Bernardy. 2024. Learning structure-aware representations of dependent types. In Advances in Neural Information Processing Systems 37 (NeurIPS 2024).

Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 521–528, Manchester, UK. Coling 2008 Organizing Committee.

Per Martin-Löf. 1984. Intuitionistic type theory. Bibliopolis.

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.

Mai Matsubara, Yasunori Hokazono, Mitsuhiro Tsunoda, Koutaro Tamura, Hinari Daido, Asa Tomita, and Daisuke Bekki. 2026. Gengogakuteki-paipurain-ni motozuku datōsei-no kenshō: Kin'yū-tekisuto-o taishō-to-shite (trans. Validity verification based on a linguistic pipeline: A case study of financial texts). In Proceedings of the 32nd Annual Conference of the Association for Natural Language Processing.

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.

Nanako Miyagawa, Sora Tagami, and Daisuke Bekki. 2025. Toward the development of an automated theorem prover "Neural Wani" for dependent type theory (in Japanese). In Proceedings of the 39th Annual Conference of the Japanese Society for Artificial Intelligence, 3G5-GS-6-04.

Tim Rocktäschel and Sebastian Riedel. 2017. End-to-end differentiable proving. In Advances in Neural Information Processing Systems 30 (NIPS 2017).

Koharu Saeki, Asa Tomita, Mai Matsubara, and Daisuke Bekki. 2026. Yesod-ni-yoru nihongo-suiron-shisutemu lightblue-no kaihatsu-kankyō-no kōchiku (trans. Building a development environment for the Japanese inference system lightblue using Yesod). In Proceedings of the 32nd Annual Conference of the Association for Natural Language Processing.

Kana Sakuma, Asa Tomita, and Daisuke Bekki. 2026. Izongata-imiron-ni-yoru nihongo-fukugō-dōshi-no imi-gōsei-to suiron (trans. Semantic composition and inference of Japanese compound verbs based on dependent type semantics). In Proceedings of the 32nd Annual Conference of the Association for Natural Language Processing.

Mark Steedman. 1996. Surface Structure and Interpretation. The MIT Press.

Asa Tomita and Daisuke Bekki. 2026. LLM-o mochiita chishiki-hokan-no tōgō-ni-yoru shizen-gengo-suiron-shisutemu lightblue-no kakuchō (trans. Extending the natural language inference system lightblue by integrating LLM-based knowledge completion). In Proceedings of the 32nd Annual Conference of the Association for Natural Language Processing.

Asa Tomita, Mai Matsubara, Hinari Daido, and Daisuke Bekki. 2025. Natural language inference with CCG parser and automated theorem prover for DTS. In Proceedings of the Second Workshop on the Bridges and Gaps between Formal and Computational Linguistics (BriGap-2), pages 1–7, Düsseldorf, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems.

Mingzhe Wang, Yihe Tang, Jian Wang, and Jia Deng. 2017. Premise selection for theorem proving by deep graph embedding. In Advances in Neural Information Processing Systems 30 (NIPS 2017).

A An Example of a Solved JSeM Problem

One of the eight problems solved by both *wani* and *Neural Wani* in our evaluation is JSeM problem 63 (GeneralizedQuantifier section):

- **Premise:** "Fukusuu no shain ga idou o kibou shite iru" ("Several employees want a transfer.")
- **Hypothesis:** "Idou o kibou shite iru shain ga iru" ("There exists an employee who wants a transfer.")
- **Answer:** yes (entailment).

For this problem, both systems construct identical proof trees of 33 rule applications (3 (ΣI), 10 (ΣE), and 20 (Membership)).

Cross-linguistic Geometry of Adjective Representations in Multilingual Transformers: Semantic Class, Gradability, and Positional Effects

Tancredi Monterosso
University of Verona / Free University of Bozen
tancredi.monterosso@univr.it

Abstract

In this study, we examine whether multilingual contextual embeddings encode properties of adjectives that are theoretically relevant to formal analyses of nominal modification. Using Universal Dependencies corpora for Arabic, English, and Italian, we extract contextualized adjective embeddings from the multilingual XLM-RoBERTa model and analyze them with respect to (i) semantic classes, (ii) the distinction between relational and descriptive adjectives, (iii) the distinction between gradable and non-gradable adjectives, and (iv) prenominal versus postnominal position in Italian. Our results indicate that adjective representations are organized in a shared multilingual space, but that this space is not best accounted for by a rigidly aligned universal hierarchy of semantic classes. Rather, the most salient organizing dimensions correspond to broader semantic-syntactic contrasts, in particular the relational/descriptive opposition, gradability, and, in the case of Italian, position-conditioned variation.

1 Introduction

Adjectives and adjectival modification have long been central topics in generative linguistics, where they provide a productive domain for investigating the internal organization of nominal modification. A substantial body of work has examined the structure of the noun phrase (NP) and has sought to account for cross-linguistic variation in the ordering and linearization of adjectives, numerals, demonstratives, and possessives (e.g., Cinque 2010 or Fassi Fehri, 1999 for Arabic). Adjective placement, in particular, varies considerably across languages: English predominantly exhibits adjective–noun (A–N) order, Arabic predominantly exhibits noun–adjective (N–A) order, and Italian productively allows both prenominal and postnominal adjectives. From a generative perspective, adjectives have been analyzed either as specifiers of functional projections within DP, with their ordering constrained by hierarchically organized semantic classes, building on distinctions such as those proposed by Dixon (1982) (e.g., Cinque 2010), or as adjectival modifiers merged freely as adjuncts, in which relative order is shaped by factors beyond a fixed cartographic hierarchy and in which externalization processes operating outside narrow syntax play a crucial role (e.g., Svenonius 2008).

Concurrently, recent work on large language models (LLMs) and multilingual contextual encoders has made it possible to investigate the internal representations learned by neural networks in much greater detail than before (Chang et al., 2022). Rather than treating such models as opaque "black boxes," current research increasingly probes whether their vector spaces encode linguistically meaningful distinctions. This line of work is relevant to generative linguistics for at least two reasons. First, it bears on long-standing theoretical questions concerning the nature of linguistic knowledge and the role of data-driven acquisition (Chesi, 2024). Second, it offers new empirical tools for evaluating whether neural representations capture distinctions that are central to formal analyses of syntax and semantics. As argued, for instance, by Kennedy (2025), neural networks may provide a useful testing ground for both generative and non-generative hypotheses about linguistic structure.

In this study, we investigate whether multilingual contextual embeddings encode theoretically relevant properties of adjectives. Using Universal Dependencies corpora for Arabic, English, and Italian, we extract contextualized adjective embeddings from the multilingual XLM-RoBERTa model. We then analyze these representations with respect to semantic classes, the relational/descriptive distinction, the gradable/non-gradable distinction, and prenominal versus postnominal adjectival position in Italian.

Our results suggest that adjective representations are organized in a shared multilingual space; how-

Third Workshop on the Bridges and Gaps between Formal and Computational Linguistics, pages 22–30

ever, this space is not best characterized by a rigidly aligned universal semantic hierarchy. Rather, the strongest organizing dimensions correspond to broader semantic-syntactic contrasts, especially relationality, gradability, and position-conditioned variation. In this respect, the findings suggest that approaches deriving adjectival order primarily from a fixed cartographic hierarchy (e.g. in Cinque, 2010) may overgenerate the observed patterns. Instead, the evidence points toward a model in which adjectival representation is shaped by factors other than semantic hierarchy.

More specifically, the Italian prenominal/postnominal contrast supports the hypothesis that adjectives may be associated with two distinct positions of external merge. On this view, surface order is not reducible to a single hierarchical template, but reflects the interaction between merge position and externalization. The neural evidence therefore suggests that externalization plays a significant role in the syntactic ordering of adjectives and should be treated as a central component of their formal analysis.

2 Data and Extraction Pipeline

2.1 Datasets and Vector Processing

We extract adjective tokens from Universal Dependencies (UD) corpora (Nivre et al., 2020) for three languages: Arabic, English, and Italian. Specifically, we use UD_Arabic-PADT for Arabic, UD_English-GUM for English, and UD_Italian-ISDT for Italian. These languages were selected because they instantiate distinct typological patterns of adjective placement: English is predominantly adjective–noun (A–N), Arabic predominantly noun–adjective (N–A), and Italian allows both prenominal and postnominal adjectives. This typological contrast makes it possible to test whether positional variation within a single language (Italian) aligns with broader cross-linguistic differences in adjectival ordering.

To control the syntactic environment as tightly as possible, we restrict extraction to adjective tokens satisfying the following criteria: UPOS = ADJ and DEPREL = amod. This restriction ensures that we focus exclusively on adjectives in noun-internal modification, excluding predicative uses and other non-attributive contexts. We also require a minimum of five occurrences per adjective lemma in order to ensure that averaged contextual embeddings reflect recurring distributional

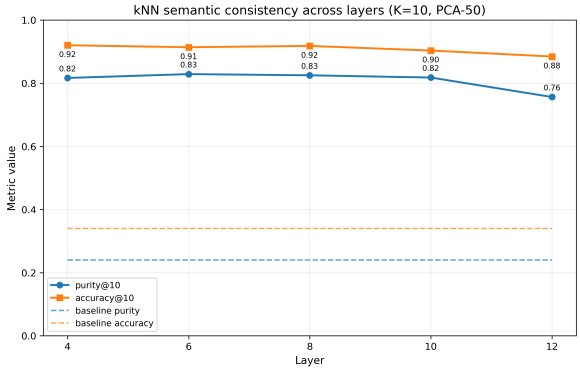


Figure 1: kNN semantic consistency across layers.

properties rather than noise arising from a small number of attested contexts. For each lemma, we aggregate all contextual occurrences and compute an averaged contextual representation (Apidianaki, 2022). Embeddings are extracted from the multilingual XLM-RoBERTa model, primarily from layer 8. This choice is motivated by a layer-wise analysis: a sweep across layers 4, 6, 8, 10, and 12 shows that semantic-class structure in local neighborhoods is most stable in the middle layers, a pattern also reported in prior work (Chang et al., 2022). In particular, kNN-based semantic consistency (which is discussed in greater detail below) remains high from layers 4 to 10 and decreases at layer 12. We therefore select layer 8 as a representative mid-layer setting rather than averaging across layers. This procedure resulted in a dataset of 1,598 adjective embeddings after applying the minimum frequency threshold of five occurrences per lemma.

Following extraction, embeddings are post-processed using mean-centering and row-wise L2 normalization, a standard procedure in representation analysis (Mahajan et al., 2025). This step reduces anisotropy and improves the interpretability of cosine similarity (Rajaee & Pilehvar, 2022). Several analyses are furthermore conducted in a PCA-reduced space (50 dimensions), which preserves a substantial portion of the original variance while providing a more stable basis for neighborhood-based diagnostics than lower-dimensional projections. In particular, we compute purity@10, that is, the average proportion of the ten nearest neighbors that share the same label as the target item, and accuracy@10, that is, the proportion of items whose label is correctly recovered by majority vote over their ten nearest neighbors. Together, these measures quantify the extent to which local neigh-

borhoods in the embedding space are homogeneous and class-consistent. To capture the relative geometry among semantic classes, we also perform Representational Similarity Analysis (RSA) (Abnar et al., 2019) using class-level centroids for each language. We additionally report a set of standard clustering evaluation measures in the Discussion section. Specifically, we compute the Silhouette score, the Calinski–Harabasz index, the Dunn index, and the Adjusted Rand Index (ARI). These measures are reported both for the original embedding space and for the PCA50-reduced space, allowing us to evaluate whether the observed clustering structure is preserved under dimensionality reduction and to verify that the main results are not an artifact of the PCA transformation.

2.2 Semantic Classes

Since Dixon (1982), adjectives have often been associated with broad semantic classes argued to constrain their ordering within the noun phrase. Within the generative tradition, and particularly in the cartographic framework developed by Cinque (2010), nominal modification is modeled as a hierarchy of functional projections whose ordering reflects Dixon's typology of adjectival classes. Under this view, cross-linguistic variation arises through movement operations affecting the noun or larger nominal constituents. If this semantic hierarchy constitutes a primary organizing principle of adjectival structure, adjectives belonging to the same class should be expected to occupy coherent regions of the embedding space. The semantic-class analysis presented below therefore investigates whether this theoretical generalization is recoverable in multilingual contextual embeddings and reflected in the geometry of adjective representations. As a starting point, we first classify adjectives into broad semantic classes using a prototype-based method. The classes considered are COLOR, SIZE, AGE, VALUE, PROP (physical property), and REL (relational). We did not include SPEED in the final analysis because the number of available items was too small to support a reliable class-level evaluation. Semantic class labels are assigned to adjective lemmas in Arabic, English, and Italian using contextual embedding representations. Class centroids are constructed from manually defined prototype adjectives, and each lemma is classified on the basis of cosine similarity to these centroids, together with thresholds on score and margin. For the COLOR class, we additionally employ a closed lexical list in order to override centroid-based classification and ensure higher precision. While classes such as AGE, VALUE, PROP, and SIZE exhibit greater semantic variability and fuzzy boundaries, color adjectives are generally characterized by a small set of lexically identifiable items whose membership is largely uncontroversial across languages. For this reason, we used a manually curated lexical list for COLOR and allowed this classification to override the centroid-based procedure applied to the other classes. The purpose of this choice was to maximize precision and avoid the introduction of classification errors arising from prototype similarity. In other words, the departure from the general procedure reflects a methodological decision motivated by the exceptional lexical stability and semantic coherence of color terms, rather than a theoretical distinction between COLOR and the other semantic classes. This procedure resulted in a total of 879 adjective embeddings after filtering (EN = 165, AR = 189, IT_PRE = 137, IT_POST = 388). Their distribution across semantic classes is reported in Figure 2. For the purposes of the semantic-class visualizations, Italian adjectives occurring in both prenominal and postnominal positions were averaged and treated as a single lexical item, thereby preventing duplicate representations of the same lemma in the plots.

Figure 2: Adjective class coverage divided by language and semantic class. Italian is divided in prenominal adjective (IT_PRE) and post-nominal adjective (IT_POST).

The resulting semantic-class structure shows that multilingual adjective vectors are organized more strongly by semantic value than by language identity. This finding is consistent with recent work on cross-linguistic representational spaces in multilingual transformers (Chang et al., 2022). PCA-based visualizations suggest that adjectives from different languages are substantially intermingled,

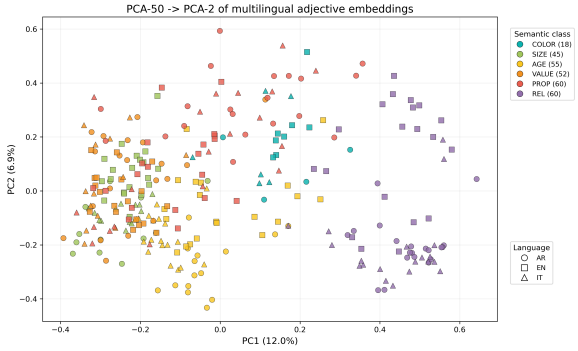


Figure 3: PCA of multilingual adjective embeddings divided by semantic class. The first two principal components explain 12.0% and 6.9% of the variance, respectively.

while still exhibiting partial grouping by semantic class, especially in the case of relational adjectives. The kNN results further support this interpretation. Overall semantic-class purity is high (Purity@10 = 0.8498) relative to the chance baseline (0.2305). The three languages display comparable values: Arabic = 0.8529 (n = 191), English = 0.8275 (n = 167), and Italian = 0.8607 (n = 285). At the class level, all categories are well above baseline, though to different degrees. REL achieves the highest absolute purity (0.94), indicating that relational adjectives form a particularly coherent cluster in the embedding space. AGE (Purity@10 = 0.84) and PROP (Purity@10 = 0.83) also exhibit strong cohesion, whereas VALUE (Purity@10 = 0.67) is comparatively more diffuse, consistent with its broader and more context-dependent semantics. Interestingly, classes with lower baseline probabilities (i.e. classes that are relatively infrequent), such as COLOR (Purity@10 = 0.78) and SIZE (Purity@10 = 0.80), display especially large purity ratios, suggesting that relatively small classes can nonetheless form well-defined regions in the embedding space when sufficient contextual evidence is available. This pattern supports the view that semantic-class structure is not merely an artifact of frequency or class imbalance.

2.3 RSA of class geometry

We use RSA in the semantic-class analysis because simple centroid alignment is not sufficient to describe how classes are organized within each language. Figure 3 can show whether, for example, Italian AGE is close to English AGE, but it does not tell us whether the overall system of relations among AGE, VALUE, REL, PROP, and the other classes is similarly structured across languages. RSA is useful precisely because it shifts the focus from isolated class-to-class comparisons to the internal geometry of the whole semantic space. In practical terms, RSA starts from the class centroids computed for each language. For each pair of semantic classes within a language, we calculate a distance value, here based on cosine distance, and arrange all these pairwise values into a class-distance matrix. This matrix summarizes how close or distant the classes are from one another in that language. We then compare these matrices across languages, typically by correlating their off-diagonal values. If two languages have similar matrices, this means that their semantic classes are organized in a similar relative geometry, even if the absolute positions of the vectors are not identical. The RSA distance matrices show a broadly comparable class geometry across the three languages, but with some clear language-specific differences. In Arabic, the largest distances involve REL, especially in its relation to SIZE, and, to a lesser extent, COLOR and AGE. COLOR is also relatively distant from SIZE and AGE, whereas PROP tends to remain comparatively closer to the other classes overall. This suggests that the Arabic class space is structured around a strong separation between REL and part of the descriptive domain, while PROP occupies a more central position.

In English, the most pronounced distances are concentrated around COLOR and REL. In particular, COLOR shows very large distances from VALUE and SIZE, and REL is also strongly separated from VALUE and SIZE. By contrast, PROP and AGE remain relatively closer to the rest of the system. Compared with Arabic, English therefore shows a sharper polarization involving COLOR and REL as more isolated regions of the class space.

Italian is globally very similar to English. As in English, the largest distances involve COLOR and REL, especially in relation to VALUE and SIZE. REL and SIZE are also highly distant, and COLOR remains strongly separated from VALUE. At the same time, PROP and AGE again occupy more intermediate positions, with lower distances to the other classes. The Italian matrix therefore largely reproduces the English configuration, with only minor local differences in magnitude.

Taken together, the matrices indicate that English and Italian are more similar to each other in their class geometry, while Arabic shows a somewhat different pattern, especially in the distribution

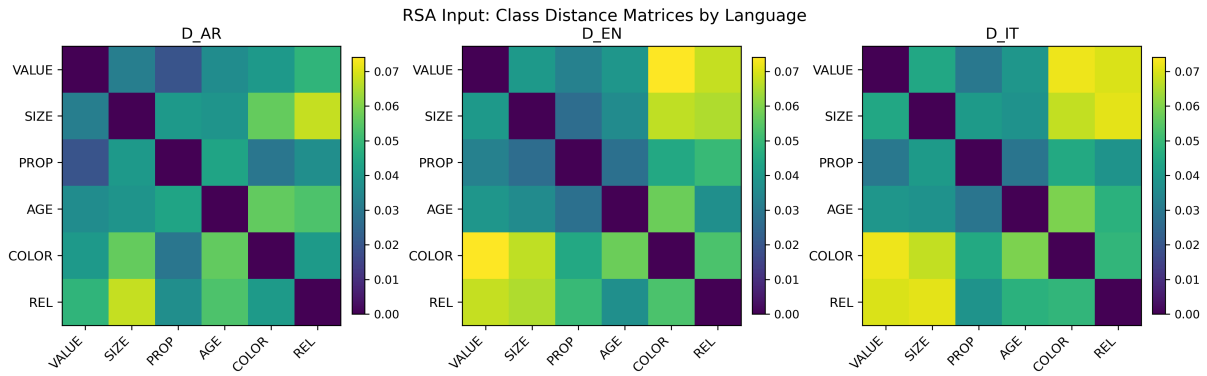


Figure 4: RSA of class distance matrices.

of distances involving REL. Thus, RSA is informative here because it reveals not only that the same classes are present across languages, but also how their relative organization differs in a measurable way from one language to another.

2.4 REL vs. DESC

The semantic-class analysis reveals that relational adjectives form the most coherent category in the embedding space, consistently yielding higher purity values than the other classes. This observation is theoretically significant, since the distinction between relational and descriptive adjectives has long been recognized as a fundamental dimension of adjectival semantics and syntax. Relational adjectives are commonly associated with classificatory or non-intersective modification, whereas descriptive adjectives denote properties that can be directly attributed to the noun (ten Hacken, 2019). Because this opposition cuts across individual semantic classes and has been argued to play a role in adjectival ordering (e.g., Cinque 2010), it provides a natural candidate for a broader organizing principle of adjective representations. We therefore collapse the semantic classes into a binary contrast between relational (REL) and descriptive (DESC) adjectives in order to test whether this distinction explains the geometry of the multilingual embedding space more effectively than the finer-grained semantic categories considered above.

The binary distinction between relational and descriptive adjectives is strongly encoded in the embedding space across all three languages. Both purity@10 and majority-vote accuracy@10 reach near-ceiling values: accuracy exceeds 0.98 in all cases, while purity remains above 0.94. These results indicate that local neighborhoods are highly homogeneous with respect to the REL/DESC distinction and that nearest neighbors overwhelmingly preserve this binary opposition. More generally, they suggest that broad structural distinctions between classifying and descriptive modification play a central role in organizing adjective representations in multilingual transformers, even more so than fine-grained semantic classes.

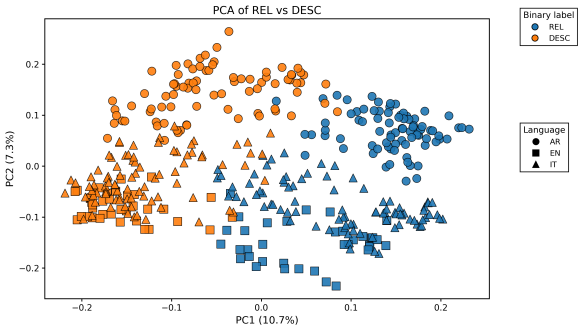


Figure 5: PCA of multilingual adjective embeddings divided into REL and DESC categories.

2.5 Gradable vs. Non-Gradable

While the REL/DESC distinction captures an important opposition between classificatory and descriptive modification, another property that has long been argued to play a central role in adjectival semantics is gradability. Since the work of Kennedy and McNally (2005), gradability has been treated as a fundamental dimension of variation among adjectives, distinguishing predicates that can participate in degree constructions and scalar interpretations from those that cannot. Beyond its semantic relevance, gradability has also been argued to interact with the syntactic behavior of adjectives, including their distribution and ordering within the nominal domain. This makes it a particularly interesting candidate for an organizing principle of multilingual adjective representa-

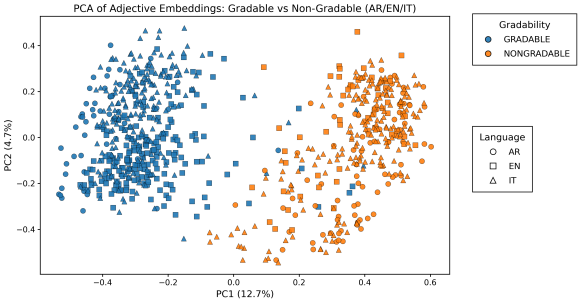


Figure 6: PCA of multilingual adjective embeddings divided into gradable and non-gradable categories.

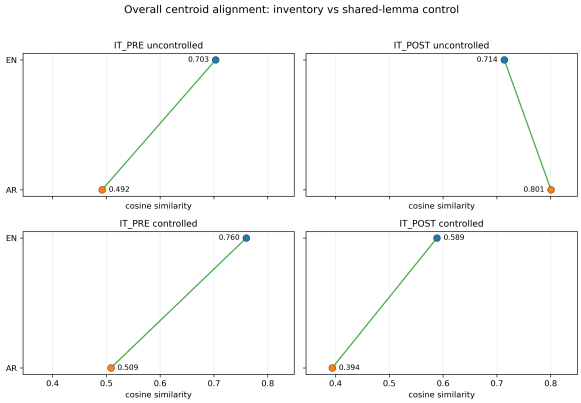


Figure 7: Overall centroid alignment between Italian (PRE/POST) and Arabic and English.

tions. We therefore investigate whether gradable and non-gradable adjectives form distinct regions of the embedding space and whether this distinction provides a stronger account of its geometry than the semantic-class and REL/DESC contrasts considered above. To derive gradability labels, we combine weak supervision from corpus-based degree-modifier evidence with a supervised classifier trained on contextual embeddings. Degree-attested adjectives are treated as positive examples, whereas relational adjectives are used as conservative negative examples. A PCA-50 plus logistic regression pipeline is evaluated by cross-validation and then fit on the full training set. The resulting model assigns each lemma a probability of being gradable; this probability is combined with thresholding and prior evidence to partition the lexicon into GRADABLE, NON_GRADABLE, and UNKNOWN sets. In this way, gradability is operationalized as a distributional property of the embedding space while preserving a conservative distinction between attested evidence and model-based generalization.

The gradable vs. non-gradable distinction is also strongly encoded in the embedding space. Across all languages, both purity@10 and accuracy@10 reach near-ceiling values (approximately 0.97–0.99), substantially exceeding their baselines. This indicates that local neighborhoods are highly homogeneous with respect to gradability and that nearest neighbors reliably preserve this distinction. The effect is robust across Arabic, English, and Italian, suggesting that gradability constitutes a stable and cross-linguistically consistent organizing dimension in multilingual contextual representations.

2.6 Positional Split in Italian

Italian provides a particularly informative internal comparison because adjectives can appear both as prenominal modifiers and postnominal modifiers. We therefore extract two separate Italian spaces: IT_PRE (prenominal adjective embeddings) and IT_POST (postnominal adjective embeddings). We compare these two spaces to English and Arabic at three levels: overall centroid alignment, class-centroid alignment, and RSA of class geometry. We also distinguish between uncontrolled analyses, which use all available lemmas, and shared-lemma controls, which restrict the comparison to lemmas attested in both Italian positions.

These analyses reveal a systematic positional effect in the embedding space. At the global centroid level, prenominal adjectives align more closely with English, while postnominal adjectives appear more similar to Arabic. However, once lexical inventory is controlled for by restricting the analysis to lemmas attested in both positions, the apparent Arabic alignment of postnominal adjectives is substantially reduced. This indicates that part of the effect is driven by distributional differences in the types of adjectives that occur in postnominal position. As shown in Figure 7, prenominal Italian exhibits consistent alignment with English in both uncontrolled and shared-lemma conditions. By contrast, postnominal Italian appears closer to Arabic only in the uncontrolled setting; this effect weakens or disappears under lexical control, suggesting that it is driven primarily by lexical inventory rather than position alone. Nevertheless, class-level centroid analyses show that the relative organization of semantic classes remains more English-like in IT_PRE and more Arabic-like in IT_POST, indicating that positional effects are not purely compositional.

3 Discussion

Taken together, the results suggest that multilingual transformer models encode adjectives in a way that is structurally meaningful, but not reducible to a simple one-dimensional universal hierarchy. Instead, the most stable dimensions across languages appear to be broad contrasts such as relational/descriptive adjectives and gradable/non-gradable adjectives, with the latter yielding the strongest results in Purity@10 and Accuracy@10. In addition to these measures, the clustering metrics further support this pattern, particularly the Adjusted Rand Index (ARI), which confirms the hierarchy observed in the previous analyses: the gradable/non-gradable distinction exhibits the strongest clustering structure, followed by the REL/DESC distinction and, finally, by semantic classes. Furthermore, the PCA reduction appears to function primarily as a denoising procedure, improving cluster compactness and separability without artificially creating the observed clustering structure.

From a generative perspective, these findings are compatible with approaches to the human faculty of language and the syntactic computational system in which adjectival modification is not reduced to a simple sequence of hierarchically ordered lexical classes, but is instead shaped by broader semantic-syntactic contrasts (e.g., Manzini 2025; Svenonius 2008). More specifically, our results suggest that semantic class alone is not the primary organizing principle in the multilingual embedding space, even though semantic classes remain recoverable and empirically useful. Rather, broader distinctions such as the relational/descriptive contrast and, even more strongly, gradability appear to structure the space more robustly.

The relational/descriptive opposition reflects a distinction that has long been recognized in the literature on adjective semantics, including contrasts related to subsective and intersective interpretation. In broad terms, descriptive adjectives tend to pattern with the former, while relational adjectives tend to pattern with the latter. As argued by Cinque (2010), this distinction is relevant to adjective ordering and should therefore be taken seriously in any formal account of nominal modification. Our results support this view by showing that the REL/DESC contrast is encoded more strongly than fine-grained semantic classes in the multilingual representation space.

Gradability, which yields the strongest results in our study, has likewise been identified as a central property of adjectival meaning, especially since Kennedy & McNally (2005), where it is treated as a key dimension of semantic variation across natural languages. The fact that the multilingual encoder

Label Set	Lang.	Dimensionality	Silh.	CH	Dunn	ARI
Semantic classes	AR	768D	0.099	9.627	0.229	0.750
Semantic classes	AR	PCA50	0.186	13.779	0.259	0.420
Semantic classes	EN	768D	0.106	6.922	0.241	0.593
Semantic classes	EN	PCA50	0.156	10.090	0.211	0.571
Semantic classes	IT	768D	0.102	10.114	0.166	0.287
Semantic classes	IT	PCA50	0.124	14.666	0.159	0.423
REL/DESC	AR	768D	0.192	24.964	0.215	0.937
REL/DESC	AR	PCA50	0.270	37.302	0.376	0.937
REL/DESC	EN	768D	0.106	11.711	0.298	0.474
REL/DESC	EN	PCA50	0.153	16.547	0.270	0.596
REL/DESC	IT	768D	0.134	24.311	0.166	0.492
REL/DESC	IT	PCA50	0.193	35.004	0.154	0.805
Gradability	AR	768D	0.346	28.812	0.393	0.966
Gradability	AR	PCA50	0.355	33.561	0.527	0.966
Gradability	EN	768D	0.131	12.103	0.296	0.741
Gradability	EN	PCA50	0.163	16.462	0.220	0.718
Gradability	IT	768D	0.145	27.760	0.105	0.778
Gradability	IT	PCA50	0.193	39.890	0.047	0.767

Table 1: Clustering metrics for adjective representations in the original and PCA50 embedding spaces. Silh. = Silhouette score; CH = Calinski–Harabasz score; Dunn = Dunn index; ARI = Adjusted Rand Index.

learns to separate and cluster adjective embeddings so effectively on the basis of gradability suggests that this feature plays a major role in the organization of adjective meaning. Moreover, because these embeddings are extracted from an intermediate transformer layer that is plausibly sensitive to both semantic and syntactic information (Jawahar et al., 2019), the results also suggest that gradability may be relevant not only to semantic interpretation, but also to the factors that shape adjective ordering. In this sense, gradability emerges as a plausible candidate for a feature that contributes to the computational organization of adjectival modification, rather than as a merely lexical or descriptive property.

The Italian PRE/POST contrast is particularly informative from a theoretical perspective. Prenominal Italian aligns more closely with English, whereas postnominal Italian shows a stronger affinity with Arabic, especially at the level of class geometry. At the same time, the shared-lemma control demonstrates that this pattern cannot be reduced to a uniform positional shift affecting the same lexical items across contexts; lexical distribution plays a substantial role. This suggests that adjective position in Italian may correspond to more than one type of modification relation, with prenominal adjectives aligning more closely with the English prenominal system and postnominal adjectives only partially resembling the Arabic postnominal system.

More broadly, these findings are consistent with the view that adjective position is not determined solely by linearization through a fixed functional hierarchy, but also reflects differences in the semantic-syntactic organization of modification (Manzini 2025). In this respect, our results are compatible with Manzini's proposal that adjectival modifiers may be externally merged in two distinct structural positions within the nominal domain: at the edge of the NP phase in the case of gradable adjectives, and in the complement domain of nP in the case of non-gradable adjectives. Although our findings on their own do not provide conclusive evidence for such a derivational account, they are consistent with an analysis in which positional alternations in Italian reflect structurally differentiated modification sites rather than a single uniform ordering mechanism.

4 Conclusion

We have presented a cross-linguistic study of adjective embeddings in Arabic, English, and Italian using multilingual transformer representations extracted from Universal Dependencies corpora. Our analyses show that adjective embeddings form a multilingual space structured by semantic classes, but even more strongly by the broader distinctions between relational and descriptive adjectives and between gradable and non-gradable adjectives, with the latter being the strongest shaping factor. We further show that adjective position in Italian conditions cross-linguistic alignment in systematic ways, with prenominal contexts behaving more similarly to English and postnominal contexts exhibiting a more Arabic-like class geometry, although this latter effect is partly driven by lexical inventory.

The present study has a number of limitations. First, the analysis is based on contextual embeddings derived from a single multilingual encoder and a restricted set of UD corpora, so the results may partly reflect model-specific and corpus-specific properties rather than general characteristics of multilingual representations. Second, our semantic-class labels are obtained through a prototype-based procedure operating on the same embedding space that is subsequently analyzed. Consequently, the semantic-class and, to a lesser extent, the REL/DESC results cannot be regarded as entirely independent of the labeling process itself and should therefore be interpreted primarily as measures of the internal coherence of the induced categories rather than as fully independent validations of their existence. Importantly, however, the strongest effects reported in this study concern gradability, whose labels are derived through a substantially different procedure combining corpus-based evidence from degree modification with weakly supervised classification. Unlike semantic classes and REL/DESC, gradability is therefore not directly induced from the same embedding-space geometry used in the subsequent analyses. The fact that gradability consistently emerges as the strongest organizing dimension across kNN, clustering, and classification measures suggests that its role cannot be reduced to an artifact of the labeling procedure. More generally, the geometric patterns observed in the embedding space should not be interpreted as direct evidence for a particular syntactic derivation, but rather as indirect represen-

tational tendencies that may inform formal analyses of adjectival modification and ordering. Future work should therefore test the robustness of these findings across additional models, corpora, and annotation procedures, and should relate the present vector-space evidence more directly to controlled syntactic data. A further limitation is that some semantic classes, particularly COLOR and SIZE, are represented by relatively few items. Consequently, class-specific results for these categories are less stable and should be interpreted with caution. Overall, these findings contribute to our understanding of what multilingual encoders represent about adjectives and adjectival modification. More broadly, they suggest that vector-space diagnostics can provide useful evidence for formal analyses of linear ordering and modification, particularly if the human syntactic computational system is viewed as an efficient computational system (Kennedy, 2025).

References

Abnar, S., Beinborn, L., Choenni, R., Zuidema, W. (2019). Blackbox Meets Blackbox: Representational Similarity & Stability Analysis of Neural Language Models and Brains. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 191–203. Florence: Association for Computational Linguistics.

Apidianaki, M. (2022). From Word Types to Tokens and Back: A survey of approaches to word meaning representation and interpretation. *Computational Linguistics*, pp. 1–59.

Chang, T. A., Tu, Z., Bergen, B. K. (2022). The geometry of multilingual language model representations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 119–136. Abu Dhabi: Association for Computational Linguistics.

Cinque, G. (2010). *The syntax of adjectives: A comparative study*. Cambridge, MA: MIT Press.

Chesi, C. (2024). Is it the end of (generative) linguistics as we know it?. *Italian Journal of Linguistics*, 37.1, pp. 3-44.

Dixon, R. M. W. (1982). *Where have all the adjectives gone? And other essays in semantics and syntax*. Berlin: Mouton.

Fassi Fehri, A. (1999). Arabic Modifying Adjectives and DP Structures. *Studia Linguistica, 53*.

Jawahar, G., Sagot, B., & Seddah, D. (2019). What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3651–3657. Florence: Association for Computational Linguistics.

Kennedy, C., McNally, L. (2005). Scale structure, degree modification, and the semantics of gradable predicates. *Language*, 81(2), pp. 345–381.

Kennedy, M. (2025). Evidence of generative syntax in LLMs. In *Proceedings of the 29th Conference on Computational Natural Language Learning*, pp. 377–396. Vienna, Austria: Association for Computational Linguistics.

Manzini, M. R. (2025). Left–right asymmetries in Italian adjectives: Partial order and phases. *Syntactic Theory and Research, 1*, pp. 1–39.

Nivre, J., de Marneffe, M. C., Ginter, F., Hajič, J., Manning, C. D., Pyysalo, S., Schuster, S., Tyers, F., & Zeman, D. (2020). Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 4034–4043. Marseille, France: European Language Resources Association.

Rajaee, S., & Pilehvar, M. T. (2022). An isotropy analysis in the multilingual BERT embedding space. In *Findings of the Association for Computational Linguistics: ACL 2022*. Dublin, Ireland: Association for Computational Linguistics.

Svenonius, P. (2008). The position of adjectives and other phrasal modifiers in the decomposition of DP. In L. McNally & C. Kennedy (Eds.), *Adjectives and adverbs: Syntax, semantics, and discourse* (pp. 16–42). Oxford: Oxford University Press.

ten Hacken, P. (2025). Relational adjectives between syntax and morphology. *SKASE Journal of Translation and Interpretation, 19(1)*, pp. 77–92. Vienna, Austria: Association for Computational Linguistics.

Diagnosing Compositional Generalization in Transformers on ReCOGS with Compositional Graph Similarity

Bruno Leite Franco and Edson Emilio Scalabrin
Graduate Program in Informatics, Pontifícia Universidade Católica do Paraná, Brazil
leite.franco@pucpr.edu.br

Abstract

This paper investigates the evaluation of compositional generalization in Transformer models on the ReCOGS benchmark. ReCOGS is a controlled synthetic benchmark designed to test whether models can systematically generalize from familiar linguistic elements to novel grammatical and semantic combinations. However, ReCOGS relies on Semantic Exact Match (SEM), a binary metric that assigns the same penalty to minor local mismatches and severe structural errors, limiting diagnostic interpretation. To address this limitation, this study introduces *Compositional Graph Similarity* (CGS), a graph-based metric that compares predicted and reference semantic structures through explicit edit operations, providing graded and interpretable structural evaluation. Controlled synthetic datasets are also used to test whether low-scoring ReCOGS categories reflect true model limitations or weaknesses in dataset coverage. Empirical results show that CGS identifies the lowest-scoring ReCOGS categories as *cp recursion* (45.0%), *obj pp to subj pp* (65.4%), and *prim to inf arg* (66.7%). Follow-up experiments show 0% SEM under depth extrapolation and constituent-role relocation, but 99.9% SEM for *prim to inf arg* in isolation. These findings support the conclusion that CGS is more informative than SEM and that Transformer limitations in ReCOGS are partly structural and partly due to dataset distribution.

1 Introduction

Compositionality is a central notion in linguistic theory and in the study of meaning representation, which states that the meaning of a complex expression arises from the meanings of its parts and from the way they are structurally combined. This property is what allows speakers to understand and produce an unbounded number of expressions, including many they have never encountered before, by systematically interpreting familiar elements in new combinations (Fodor and Pylyshyn, 1988). This principle is important for natural language processing because language understanding requires more than sensitivity to lexical co-occurrence. When evaluating Transformers and large language models, this issue has become even more relevant, as strong empirical performance often coexists with uncertainty about whether models are learning transferable compositional principles or merely exploiting statistical regularities present in training data (Lake and Baroni, 2018).

To investigate this question under controlled conditions, ReCOGS (*Revised COGS*) (Wu et al., 2023) provides a synthetic benchmark specifically designed to evaluate compositional generalization. It isolates linguistically meaningful variation in grammatical structure from unintended biases found in natural corpora. In this setting, models must generalize across systematic combinations rather than rely only on memorized surface patterns. However, the benchmark uses exact semantic correspondence between prediction and reference as its standard evaluation metric. This choice causes it to penalize qualitatively different errors equally. As a result, minor non-structural deviations and severe structural violations cannot be distinguished.

This limitation motivates the search for an evaluation metric that is better aligned with the structural nature of the task. The present work introduces Compositional Graph Similarity (CGS), a metric designed for ReCOGS that distinguishes structural from non-structural deviations and yields more interpretable diagnostic results. CGS is designed for synthetic datasets like ReCOGS, that require evaluation methods that are sensitive to logical form structure.

2 Background

2.1 ReCOGS

ReCOGS is a revised version of COGS (Kim and Linzen, 2020) designed to preserve the original

Third Workshop on the Bridges and Gaps between Formal and Computational Linguistics, pages 31–39

benchmark's semantic goals while reducing artifacts introduced by the target logical-form notation. It is constructed by applying a set of meaning-preserving changes to the original task. ReCOGS standardizes aspects of the representation by treating proper nouns more uniformly and by separating events from semantic roles in a Neo-Davidsonian style. These revisions preserve the original evaluation partition sizes while expanding the training set from 24,155 instances in COGS to 135,547 in ReCOGS (Wu et al., 2023).

The benchmark continues to target compositional generalization, but some generalization classes are especially relevant for structural analysis. In lexical-role shifts such as *Subj → Obj Proper*, *Prim → Subj Proper*, and *Prim → Obj Proper*, the model must assign a familiar item to a grammatical role not observed during training. In structural splits, the challenge is more directly compositional. In *Obj PP → Subj PP*, the model is trained on object-modifying prepositional phrases and tested on subject-modifying ones. In *CP Recursion*, test examples contain deeper clausal embedding than training examples. In *PP Recursion*, the model must interpret recursively stacked nominal modifiers of greater depth than those seen during training.

For illustration, the examples below are shown in simplified ReCOGS-style logical:

- **Obj PP → Subj PP**

  Train: Emma ate **the cake on the table**.
  `Emma(15); cake(23); table(41); eat(11) AND agent(11,15) AND theme(11,23) AND nmod.on(23,41)`

  Generalization: **The cake on the table** burned.
  `cake(23); table(41); burn(58) AND theme(58,23) AND nmod.on(23,41)`

- **CP Recursion**

  Train: Noah knew that Emma said that the cat painted.
  `Noah(26); cat(33); know(10); Emma(17) AND agent(10,26) AND ccomp(10,21) AND say(21) AND agent(21,17) AND ccomp(21,44) AND paint(44) AND agent(44,33)`

  Generalization: Noah knew that Emma said that **John saw that** the cat painted.
  `Noah(26); cat(33); know(10); Emma(17); John(28) AND agent(10,26) AND ccomp(10,21) AND say(21) AND agent(21,17) AND ccomp(21,52) AND see(52) AND agent(52,28) AND ccomp(52,44) AND paint(44) AND agent(44,33)`

- **PP Recursion**

  Train: John saw the ball in the bottle in the box.
  `ball(14); bottle(29); box(37); see(5) AND agent(5,John) AND theme(5,14) AND nmod.in(14,29) AND nmod.in(29,37)`

  Generalization: John saw the ball in the bottle in the box **on the floor**.
  `John(19); ball(14); bottle(29); box(37); floor(48); see(5) AND agent(5,19) AND theme(5,14) AND nmod.in(14,29) AND nmod.in(29,37) AND nmod.on(37,48)`

In these examples, the numbers in parentheses are variable identifiers, not numerical values or part of the lexical meaning. For instance, in `Emma(15)`, the identifier 15 denotes the entity labeled `Emma`; in `eat(11)`, the identifier 11 denotes an event labeled `eat`. Relations such as `agent(11,15)` and `theme(11,23)` then specify how these variables are connected: variable 15 is the agent of event 11, and variable 23 is its theme. The specific numbers assigned to variables are arbitrary and may be freely renamed, provided that the same identifier continues to refer to the same entity or event throughout the logical form and that distinct variables preserve the intended semantic distinctions. Thus, two logical forms that differ only by a consistent renaming of variable identifiers are semantically equivalent.

Empirically, ReCOGS remains challenging even after these revisions. In the original report, both LSTM and Transformer models still achieve near-100% performance on the IID test split, but structural generalization remains difficult. For the *Obj PP → Subj PP* split, the reported average Semantic Exact Match is 13.4% for LSTMs and 4.5% for Transformers, and performance on recursive structural splits remains in the single-digit to low-double-digit range. At the same time, lexical splits are often substantially easier, which indicates that structural recombination remains the more difficult component of the benchmark (Wu et al., 2023).

The canonical evaluation metric for ReCOGS is *Semantic Exact Match* (SEM). Under SEM, a prediction is counted as correct only if there exists a semantically consistent conversion of the variables in the predicted logical form into those of the reference. Operationally, this amounts to checking whether there is a bijection between predicted and gold variables such that corresponding variables participate in exactly the same predications; conjunctions are treated as an order-free set, and exact match is computed as set equality after variable conversion. This makes SEM more adequate than raw string matching, since it ignores irrelevant differences in conjunct order and variable names (Wu et al., 2023). However, it remains a binary metric, therefore if exact equivalence is not satisfied, all errors receive the same score. As a result, minor local deviations and severe structural failures are collapsed into the same outcome, and the metric does not transparently reveal where the semantic structure broke down. For a synthetic benchmark such as ReCOGS, whose examples may contain substantial lexical overlap while differing crucially in grammatical organization, this lack of structural granularity limits the interpretability of the evaluation.

2.2 Quality Criteria for Graph Similarity in ReCOGS

To compare graph-based semantic evaluation metrics in a principled way, this work adopts seven quality criteria for graph similarity proposed by (Opitz et al., 2020). The first is *continuous range*, meaning that the metric should map graph pairs to a continuous interval in $[0, 1]$, where 1 denotes maximal similarity and 0 minimal similarity. The second is *identity of indiscernibles*: a score of 1 should be assigned if and only if the compared graphs are identical under the intended notion of equivalence. The third is *symmetry*, so that comparing prediction to reference yields the same value as comparing reference to prediction. The fourth is *determinism*, requiring fixed inputs to always produce the same output or within a negligible error ϵ. The fifth is *absence of unjustified bias*, that is, the metric should not overvalue some graph regions or substructures in an undocumented or unwanted way.

The sixth criterion is *symbolic correspondence*, here understood as monotonicity with respect to symbolic overlap: if one graph shares more correct symbolic structure with the reference than another, its score should not be lower. The seventh is *graded correspondence*, namely the ability to assign partial credit when the prediction is close to the reference without being fully correct. In many semantic graph benchmarks, this seventh criterion is operationalized through soft lexical matching, for instance by using cosine similarity between node embeddings. However, this strategy is not appropriate for ReCOGS. The benchmark was designed with tightly controlled lexical choices and with emphasis on grammatical recombination rather than graded lexical relatedness. As a result, node labels are intended to function as discrete semantic symbols, not as points in a lexical similarity space. Introducing cosine-based partial matching would therefore inject an external notion of similarity that is not part of the benchmark's design and could blur precisely the distinction that ReCOGS is meant to test.

Under these criteria, *Semantic Exact Match* (SEM), the canonical ReCOGS metric, satisfies only part of what is desirable from a graph similarity measure. SEM only satisfies *identity*, *symmetry*, *determinism*, and *absence of unjustified bias*, because its decision rule is binary, order-independent under the allowed normalization. Nevertheless, SEM fails the *continuous range*, *symbolic correspondence* and *graded correspondence*, because it provides no partial credit and therefore cannot distinguish small local mismatches from deep structural errors.

Beyond Semantic Exact Match, other semantic graph metrics have been proposed in the literature, most notably *SMATCH* (Cai and Knight, 2013), *SemBLEU* (Song and Gildea, 2019), and *S^2MATCH* (Opitz et al., 2020). However, none of them simultaneously satisfy all seven quality criteria adopted in this work while also remaining fully appropriate for ReCOGS. According to the comparison used in this study, SMATCH satisfies continuous range, symbolic correspondence, and, up to negligible numerical variation, identity, symmetry, and determinism; it is also applicable to ReCOGS, but it does not provide graded correspondence. SemBLEU is likewise applicable to ReCOGS and has a continuous range, but it fails identity and symmetry and introduces an undesirable bias toward high-degree or central nodes, since its path-based formulation overweights errors in structurally prominent regions. S^2MATCH comes closer to a graded metric by replacing exact concept matching with soft similarity, but this graded

correspondence operates only at the lexical level. Because ReCOGS is a synthetic benchmark whose labels are intended to behave as discrete grammatical symbols rather than items in a lexical similarity space, such soft lexical matching is not appropriate in this setting. For this reason, although these metrics each capture part of what is desirable in structural evaluation, none provides the combination of formal adequacy, graded structural comparison, and ReCOGS compatibility required here.

These limitations motivate the search for semantic graph metrics that go beyond binary correctness. Existing alternatives, such as *SMATCH*, *SemBLEU*, and *S²MATCH*, partially address this need by introducing graded comparison between predicted and reference structures. However, none of them both satisfies all seven quality criteria and remains fully suitable for ReCOGS. Although they improve on SEM in certain respects, they either lack some of the formal or interpretability properties required here, or depend on lexical soft matching that is inappropriate for a synthetic benchmark whose labels are meant to operate as discrete grammatical symbols. Consequently, the need remains for a metric that supports graded structural evaluation while respecting the design principles of ReCOGS.

3 Experiment 1

This experiment evaluates whether a graph-based structural metric can provide a more informative diagnosis of compositional generalization in ReCOGS than the benchmark's canonical binary evaluation. The empirical analysis is based on predictions from the Transformer baseline reported in the original ReCOGS study. This baseline was an encoder–decoder Transformer with two layers in each component, four attention heads, learnable absolute positional embeddings, and hidden size 300 (Wu et al., 2023).

The central idea is that exact semantic match is appropriate as a strict correctness criterion, but insufficient to characterize the severity of semantic deviations. To address this limitation, the experiment introduces Compositional Graph Similarity (CGS), a continuous metric defined over semantic graphs and grounded in edit operations. The experiment asks two questions: whether CGS satisfies the quality criteria required of a graph similarity measure in the ReCOGS setting, and what additional empirical insights it reveals when applied to the predictions of the ReCOGS Transformer baseline.

3.1 Method

Let each prediction and gold reference be represented as a labeled directed multigraph

$$G = (V, E, \ell_V, \ell_E), \tag{1}$$

where V is the set of nodes, E is the multiset of directed edges, ℓ_V assigns labels to nodes, and ℓ_E assigns labels to edges. This representation is appropriate for ReCOGS because semantic correctness depends on preserving argument structure, event structure, edge direction, and role labels rather than on string identity alone. Because ReCOGS is a synthetic benchmark with controlled vocabulary and no intended graded lexical similarity, partial node matching cannot be justified through cosine similarity or embedding-space proximity. Instead, label similarity must be defined in grammatical rather than lexical terms.

Given a predicted graph G_p, a reference graph G_r, and a partial node alignment

$$\pi : V_p \to V_r \cup \{\perp\}, \tag{2}$$

an *edit listing* is defined as an ordered sequence

$$\mathcal{L}(G_p, G_r; \pi) = \langle o_1, o_2, \ldots, o_k \rangle, \tag{3}$$

where each o_i is an atomic edit operation. The operation alphabet includes node insertion, node deletion, node relabeling, edge insertion, edge deletion, edge relabeling, and edge reversal:

$$\Sigma = \left\{ \begin{array}{l} \texttt{insV}(v, \ell_V), \texttt{delV}(v), \\ \texttt{relV}(v, \ell_V \to \ell'_V), \\ \texttt{ins\_E}(u, v, \ell_E),\ \texttt{delE}(u, v, \ell_E), \\ \texttt{relE}((u, v), \ell_E \to \ell'_E), \\ \texttt{revE}(u, v, \ell_E) \end{array} \right\} \tag{4}$$

Where `insV`, `delV` and `relV` denote node insertion, deletion, and relabeling, and `insE`, `delE`, `relE` and `revE` denote edge insertion, deletion, relabeling, and reversal, respectively.

Since determining minimal graph edit distance is a NP-hard problem (Blumenthal and Gamper, 2020), the alignment π is obtained heuristically in two stages by applying the Hungarian algorithm (Kuhn, 1955).

To define graded penalties for relabeling, let $\mathcal{T} = (\mathcal{C}, \prec)$ be a rooted grammatical taxonomy (Figure 1 was the taxonomy used for this study).

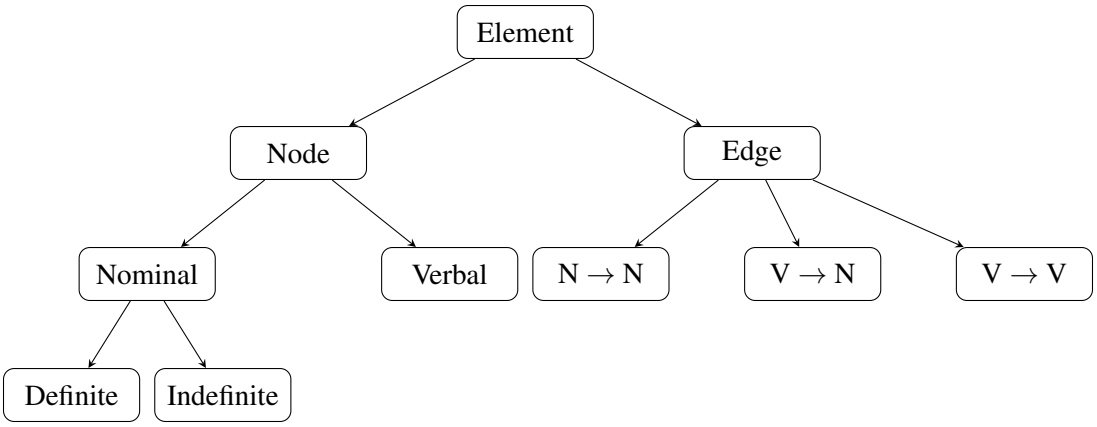


Figure 1: Taxonomy of grammatical classes for edit cost.

Node labels and edge labels are mapped to grammatical classes through a function:

$$\mathrm{cls}(\ell) \in \mathcal{C}. \tag{5}$$

Let $\mathrm{d}(c)$ be the depth of class c in the taxonomy, and let $\mathrm{a}(c_1, c_2)$ denote the lowest common ancestor of c_1 and c_2 and x and y denote class of ℓ and ℓ' respectively. The grammatical distance between labels is then defined as:

$$d_{\mathcal{T}}(\ell, \ell') = \frac{d(x) + d(y) - 2\,d(a(x,y)) + 2}{d(x) + d(y) + 2} \tag{6}$$

This definition ensures that relabelings within the same grammatical branch are penalized less than relabelings across branches, which is crucial in ReCOGS, where grammatical function matters but lexical similarity is not part of the task design.

The total grammatical edit cost is defined as

$$\mathrm{Cost}_{\mathrm{gram}}(\mathcal{L}) = \sum_{o \in \mathcal{L}} \mathrm{cost}_{\mathrm{gram}}(o), \tag{7}$$

with operation costs

$$\begin{aligned}
\mathrm{cost}_{\mathrm{gram}}(\texttt{relV}(v, \ell_V \to \ell'_V)) &= d_{\mathcal{T}}(\ell_V, \ell'_V),\\
\mathrm{cost}_{\mathrm{gram}}(\texttt{relE}((u,v), \ell_E \to \ell'_E)) &= d_{\mathcal{T}}(\ell_E, \ell'_E),\\
\mathrm{cost}_{\mathrm{gram}}(\texttt{revE}(u, v, \ell_E)) &= \rho_{\mathrm{rev}},\\
\mathrm{cost}_{\mathrm{gram}}(\texttt{insE}(u, v, \ell_E)) &= 1,\\
\mathrm{cost}_{\mathrm{gram}}(\texttt{delE}(u, v, \ell_E)) &= 1,\\
\mathrm{cost}_{\mathrm{gram}}(\texttt{insV}(v, \ell_V)) &= 1,\\
\mathrm{cost}_{\mathrm{gram}}(\texttt{delV}(v)) &= 1.
\end{aligned} \tag{8}$$

In the present formulation, edge reversal is assigned a fixed intermediate penalty ρ_{rev}, reflecting the fact that direction errors are more severe than a small within-branch relabeling but less severe than adding or deleting structure altogether.

To make scores comparable across instances of different sizes, the cost is normalized by a symmetric size term:

$$L_{\mathrm{sym}} = \min\big(|V_r| + |E_r|,\ |V_p| + |E_p|\big), \tag{9}$$

$$\mathrm{Cost}^{\mathrm{norm}}_{\mathrm{gram}}(G_p, G_r; \pi) = \frac{\mathrm{Cost}_{\mathrm{gram}}(\mathcal{L}(G_p, G_r; \pi))}{\max(L_{\mathrm{sym}}, 1)}. \tag{10}$$

Finally, *Compositional Graph Similarity* is defined as

$$\mathrm{CGS}(G_p, G_r; \pi) = \exp\big(-\mathrm{Cost}^{\mathrm{norm}}_{\mathrm{gram}}(G_p, G_r; \pi)\big). \tag{11}$$

By construction,

$$\mathrm{CGS}(G_p, G_r; \pi) \in (0, 1], \tag{12}$$

with value 1 if and only if the normalized edit cost is 0, and values decreasing continuously as structural divergence increases.

Under this definition, CGS satisfies the seven quality criteria required for graph similarity in the ReCOGS setting. First, it has a continuous range in $(0, 1]$. Second, it satisfies identity of indiscernibles, since $\mathrm{CGS} = 1$ holds only when no edit is required. Third, it is symmetric, given symmetric operation costs. Fourth, it is deterministic once alignment and tie resolution are fixed. Fifth, it avoids unjustified bias because penalties are explicitly documented and shared across graph regions rather than emerging from hidden path-counting effects. Sixth, it satisfies symbolic correspondence: if a prediction shares more correct graph structure with the reference, the number and severity of edits necessarily decrease, which increases CGS. Seventh, it

Criterion	Exact Match	SMATCH	SemBLEU	S^2MATCH	CGS (Ours)
Continuous Range	✗	✓	✓	✓	✓
Identity	✓	$\checkmark_\epsilon$	✗	$\checkmark_\epsilon$	$\checkmark_\epsilon$
Symmetry	✓	$\checkmark_\epsilon$	✗	$\checkmark_\epsilon$	$\checkmark_\epsilon$
Determinacy	✓	$\checkmark_\epsilon$	✓	$\checkmark_\epsilon$	$\checkmark_\epsilon$
Low bias	✓	✓	✗	✓	✓
Symbolic Matching	✗	✓	✗	✓	✓
Graded Matching	✗	✗	✗	$\checkmark_{\text{LEX}}$	$\checkmark_{\text{GRA}}$
Applicable to ReCOGS	✓	✓	✓	✗	✓

$\checkmark_\epsilon$: Satisfies up to statistically negligible numerical error;
$\checkmark_{\text{LEX}}$: Graded matching only at the lexical level.
$\checkmark_{\text{GRA}}$: Graded matching only at the gramatical level.

Table 1: Comparison by criteria (rows) and metrics (columns) for use in ReCOGS.

supports graded correspondence through taxonomically grounded relabel penalties and structural edit costs.

3.2 Result

As presented in Table 1, CGS is an adequate semantic evaluation metric for ReCOGS. Unlike Semantic Exact Match, CGS satisfies all seven quality criteria adopted in this work. At the same time, it remains applicable in a benchmark such as ReCOGS, where node labels are not partially matched through lexical similarity.

When applied to ReCOGS predictions, CGS highlighted three test categories with particularly low structural similarity: *cp recursion* (CGS $=$ 45.0%), *obj pp to subj pp* (CGS $=$ 65.4%), and *prim to inf arg* (CGS $=$ 66.7%). This makes these categories especially informative for further investigation.

Rather than treating these low scores only as endpoint evaluation results, they were used as signals for the next stage of experimentation. The purpose of these follow-up datasets was to determine whether the observed failures were caused by the intrinsic difficulty of the target generalization itself, or instead by insufficient exposure to the relevant structural pattern in the original ReCOGS training set.

4 Experiment 2

Experiment 1 showed that the three lowest-scoring ReCOGS categories under CGS were *cp recursion*, *prim to inf arg*, and *obj pp to subj pp*. Because these categories were the ones in which structural similarity also degraded most strongly, they were selected for a second experiment based on synthetic controlled datasets. The purpose of Experiment 2 was not merely to test if these low scores are due to a model weakness or insufficient structural representation of the relevant pattern in the original training distribution. To make this distinction explicit, each of the three phenomena was tested in isolation, with datasets designed so that the target generalization was the only source of out-of-distribution pressure.

4.1 Method

Three synthetic diagnostic datasets were constructed, each corresponding to one of the categories with the lowest CGS in Experiment 1. The first targeted *cp recursion* involved training 5 models by varying the maximum supervised depth of clausal embedding (from 1 to 5) and then evaluating the model on test sets with depths ranging from 0 to 5 (even if the model was not exposed to that depth during training). The second targeted *prim to inf arg*, in which the model was trained on isolated lexical items and on sentences following the pattern `A V_1 to V_2`, was subsequently tested on examples in which those lexical items were integrated into full sentential structures.The third targeted *obj pp to subj pp* and involved the creation of seven models, each trained on a different non-empty combination of the three argument positions that could host a structurally complex nominal constituent, so that the effect of relocating such complexity across positions could be evaluated in isolation.

The synthetic datasets were generated with a template-based sentence generator designed to produce paired surface sentences and ReCOGS-style logical forms. The generator samples lexical items from WordNet (Miller, 1992). Verbs are sampled in base form and inflected to simple past for the

surface sentence, while the corresponding logical form uses the base verb. Nouns are sampled as lowercase single-token lemmas, and prepositional modifiers are generated using the prepositions *in*, *on*, and *beside*.

The generator uses controlled templates so that the source of generalization pressure can be isolated. For the primitive-to-infinitival-argument condition, the basic template follows the form $A\ V_1\ T$ *to* R, with logical forms encoding the corresponding agent, theme, and recipient relations. For the constituent-relocation condition, one nominal argument is expanded into a complex noun phrase of the form $N\ P\ C$, where P is a preposition and C is a complement noun. The complex constituent can appear in the agent, theme, or recipient position, yielding separate datasets for each possible host role. Training and test conditions are then defined by controlling which argument positions contain complex constituents during training and which positions are held out for generalization.

The construction procedure also enforces distributional controls intended to prevent trivial lexical explanations for the results. Within each generated sentence, role fillers are kept distinct, lexical items are balanced across grammatical roles as far as possible, and verb, noun, and preposition usage is made approximately uniform. The generator further avoids repeated local pairings when possible, using randomized construction with limited backtracking and only minimal relaxation of pair-uniqueness constraints when necessary. As a result, the diagnostic datasets differ primarily in the structural configuration being tested rather than in uncontrolled lexical frequency effects.

4.2 Result

As summarized in Table 2, the comparison between ReCOGS and the diagnostic datasets reveals that the three low-scoring categories selected from Experiment 1 do not reflect a single underlying source of failure. Whereas *cp recursion* and *obj pp to subj pp* remain difficult under controlled isolation, *prim to inf arg* shifts from failure in ReCOGS to near-ceiling performance in the diagnostic setting.

In the *cp recursion* dataset, models remained highly accurate within the depth regime represented in training, but failed categorically under depth extrapolation. More specifically, inlier performance was above 99% for all in-support conditions from P2 onward, whereas the shallowest inlier conditions were noticeably lower, with 78.1% Exact Match for the P0 model on test P0 and 75.4% for the P1 model on test P1. By contrast, whenever test depth exceeded the maximum depth observed during training, Semantic Exact Match dropped to 0%, independently of the model's training regime. This pattern indicates that the models learned depth-specific configurations rather than an abstract recursive rule that could generalize beyond the structural support available in training.

Category	**ReCOGS**	**DDG**
cp recursion	12.7%	0.0%
prim to inf arg	0.0%	99.9%
obj pp to subj pp	11.9%	0.0%

Table 2: Exact Match comparison between ReCOGS and our Diagnostic Datasets Generalization (DDG).

The sub-ceiling results in the shallow inlier conditions are best interpreted as a side effect of the random node-indexing scheme introduced to enable extrapolation to larger logical forms. In low-depth examples, where the number of nodes is small, the model appears to face greater difficulty abstracting away from index variability and isolating the underlying compositional structure. As a result, the approximately 0.75 inlier accuracies observed for P0 and P1 likely reflect noise induced by random indices in structurally small graphs, rather than failure on the recursive phenomenon itself.

The result for *prim to inf arg* was markedly different. In the isolated synthetic test, the model was highly successful (99.9% Semantic Exact Match), which shows that this generalization can, in fact, be learned reliably when the relevant structure is sufficiently represented during training. The discrepency of this high values with those in the ReCOGS dataset inplies a dataset weakness in the original dataset since only 6.2% of the origial dataset has the *xcomp* relationship or weakly cyclical logical forms (the rest of the dataset has acyclical logical forms).

The *obj pp to subj pp* experiment yielded a pattern closely parallel to that of *cp recursion*. In all inlier conditions, Semantic Exact Match remained above 99.8%, indicating that the models could reliably handle structurally complex noun phrases as long as those phrases appeared in the thematic positions supported during training. In contrast, performance dropped to 0% in every outlier condition, that is, whenever a structurally complex nominal constituent had to be reanchored in a thematic

position not represented as complex in the corresponding training distribution. This sharp contrast shows that the difficulty is not due merely to lexical novelty or to the presence of internal nominal modification. Rather, it arises from the relocation itself: the model fails when a constituent enriched by internal structure must be reassigned to a new grammatical role while preserving the correct verb–argument dependencies.

Taken together, these results show that low CGS can arise from different underlying causes. In *cp recursion* and *obj pp to subj pp*, the isolated experiments preserved the original pattern of difficulty, indicating genuine structural limitations in hierarchical embedding and constituent reanchoring. In *prim to inf arg*, however, the isolated experiment produced high performance, suggesting that the original failure in ReCOGS is largely due to insufficient structural representation of the relevant configuration in the training data.

5 Conclusion

This work argued that *Compositional Graph Similarity* (CGS) is a more adequate metric than *Semantic Exact Match* for evaluating compositional generalization in ReCOGS. Whereas Semantic Exact Match is restricted to a binary notion of correctness, CGS provides a graded structural comparison, satisfies the seven quality criteria adopted in this work, and remains applicable in synthetic semantic datasets where lexical soft matching is not appropriate. Because CGS is derived from explicit edit operations, it also offers a more interpretable account of model behavior, making it possible to distinguish minor local deviations from genuine structural failures.

The experiments further showed that low performance should not always be interpreted in the same way. In the *prim to inf arg* case, the isolated synthetic test yielded high accuracy, suggesting that the poor result in ReCOGS is not evidence of an inability to perform the target generalization itself, but rather of a limitation in the dataset distribution. More specifically, the relevant structural configuration appears to be underrepresented in the original benchmark. This result indicates that Transformer models are in fact capable of abstracting primitive lexical items and preserving global semantic structure when the required pattern is adequately represented during training.

By contrast, the results for *obj pp to subj pp* and *cp recursion* point to a different conclusion. In both cases, even controlled synthetic testing preserved the pattern of failure, indicating that the problem is not merely a property of the original dataset distribution. Instead, these results suggest that the models fail to abstract reusable substructures in a sufficiently systematic way. As a consequence, they struggle when required to interact with a novel structure, even when that structure is composed only of substructures that were individually familiar during training.

More broadly, this work suggests that compositional generalization should be evaluated not only by whether a final answer is correct, but also by how the predicted semantic or inferential structure differs from the target structure. In natural language inference, prior work has shown that high aggregate accuracy can mask reliance on shallow syntactic heuristics, such as lexical overlap or subsequence matching (McCoy et al., 2019). Similarly, recent reasoning benchmarks increasingly emphasize structured logical or proof-based representations, as in ProofWriter (Tafjord et al., 2021) and FOLIO (Han et al., 2024). For such tasks, a graph-based evaluation framework could complement exact-match or label-based accuracy by identifying whether an error reflects a minor local mismatch, an incorrect role assignment, a broken inference step, or a deeper failure to preserve compositional structure. Thus, the contribution of CGS is not only a metric for ReCOGS, but also an example of how future evaluations of semantic parsing, NLI, and reasoning systems can move toward more diagnostic, structure-sensitive assessment.

6 Limitations

CGS depends on design choices in graph construction, alignment, and edit costs. Although these choices were made to reflect grammatical structure and to satisfy the quality criteria adopted in this work, different taxonomies or cost schemes could lead to somewhat different similarity profiles.

To assess if the observed failures to abstract reusable substructures from transformer models are not limited to the ReCOGS dataset. Future work should apply the same combination of structural evaluation and controlled synthetic isolation to other benchmarks.

References

David B. Blumenthal and Johann Gamper. 2020. On the exact computation of the graph edit distance. *Pattern Recognition Letters*, 134:46–57. Applications of Graph-based Techniques to Pattern Recognition.

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.

Jerry A. Fodor and Zenon W. Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3–71.

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, and 16 others. 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.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

H. W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2873–2882. PMLR.

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.

George A. Miller. 1992. WordNet: A lexical database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Juri Opitz, Letitia Parcalabescu, and Anette Frank. 2020. AMR similarity metrics from principles. *Transactions of the Association for Computational Linguistics*, 8:522–538.

Linfeng Song and Daniel Gildea. 2019. SemBleu: A robust metric for AMR parsing evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4547–4552, Florence, Italy. Association for Computational Linguistics.

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.

Zhengxuan Wu, Christopher D. Manning, and Christopher Potts. 2023. ReCOGS: How incidental details of a logical form overshadow an evaluation of semantic interpretation. *Transactions of the Association for Computational Linguistics*, 11:1719–1733.

Polar Questions in SPA–TTR: Linking Dialogue, Acquisition, and Neurosemantics

Jonathan Ginzburg[1], Shiyun Dong[1], Staffan Larsson[2], Robin Cooper[2], Andy Lücking[3]
[1]Université Paris Cité, CNRS, [2]University of Gothenburg, [3] Technische Universität Chemnitz

Abstract

In this paper, we offer an extension of an earlier proposal for treating wh-questions within a compositional, neurally–implemented semantic framework that interfaces with memory to the other main class of questions, namely polar questions. Our proposal yields improved empirical coverage for polar questions as compared with previous formal semantic accounts. It also offers the basis for an account of the finding that understanding for wh-questions emerges in language development *before* that of polar questions–a finding that goes against all previous formal semantics accounts of questions where polar questions are simplest in terms of their semantic complexity.

1 Introduction

Formal semantics and cognitive neuroscience have not always been in tight collaboration. Frege (1918, e.g., p. 69) (the originator of compositionality) believed (scientific) analysis of meaning should be divorced from psychological considerations, whereas (Fodor, 1974) argues that appealing to 'lower-level' sciences is useless for understanding cognitive systems, giving rise to the criteria of (Fodor and Pylyshyn, 1988) for explanatory cognitive systems which includes compositionality.

Considerable progress in bringing semantics and cognitive neuroscience together has been achieved by work pioneered in the Semantic Pointer Architecture (SPA) (Eliasmith, 2013). This framework underpins 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, 2004; Schlegel et al., 2022).

The basic strategy of the SPA is to couple a VSA's algebraic structure on a vector space, with coding and decoding operations into ensembles of neurons. In this way, one can both maintain compositional analyses of natural language in a transparent way that differs sharply from Large Language Model approaches, while retaining the robustness and the continuity vectors provide in semantic space.

While significant progress has been made within such an approach, it has dealt so far with relatively simple semantic phenomena. (Larsson et al., 2025) offers extension of this approach to a complex semantic phenomenon involving variable binding, developing an analysis of wh–questions. In contrast to existing formal semantic treatments, this analysis enables an interface with memory, since answering questions involves 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 largest city in Kurdistan?

b. Are you aware of the tsetse fly on your plate?

In this paper, we extend this approach to polar questions. We analyze a polar question $p?$ as a propositional function $\lambda P.P(p)$ that seeks to instantiate a modifier of the proposition p and show how this can be implemented in SPA. What emerges is a unified, compositional treatment of the two major classes of questions[1] in a neurosemantic framework. Beyond the explicit interface with memory, our analysis of polar questions is superior to existing accounts of polar questions developed in the formal semantics literature in two important ways:

[1]Evidence from 10 languages show these two classes constitute typically >95% of questions (Bolden et al., 2023).

Third Workshop on the Bridges and Gaps between Formal and Computational Linguistics, pages 40–49

1. **Answerhood**: Formal semantic treatments of polar-questions (with a few exceptions discussed below) analyze these as the simplest type of question, whose answers are 'yes' and 'no' (hence the term 'yes/no–questions). As we will demonstrate, this widely held assumption is contradicted by both introspective and corpus data that shows such questions give rise to an intrinsically wider range of answers, which our analysis captures.

2. **Order of emergence in language development**: recent data from language acquisition (Moradlou et al., 2021) shows that across languages, wh-questions are understood by infants *before* polar questions, despite biases favouring the latter in the input.[2] As Moradlou et al. (2021) demonstrate, (in the languages they consider and probably in general) polar questions are clearly not more complex syntactically than wh-questions. This suggests that polar questions are in some way more complex semantically than (certain types of) wh-questions, contradicting virtually all existing formal semantic analyses, as discussed below. Our analysis underpins the explanation offered by Moradlou et al. (2021) for the order of emergence, tied as it is to the relative difficulty of grounding answers of a given type of question to elements in the context. For the early understood wh-questions, these are concrete entities (*Carer: What is this?* (Carer points to the object): *A pencil*), whereas for polar questions these are (abstract) propositional modifiers (*Carer: Is the pencil green* Yes (It is green)/No (It is not green)).

The structure of the paper is as follows: Section 2 provides background on existing work on polar questions, the Semantic Pointer Architecture (SPA), and Type Theory with Records, the high level symbolic theory we use for semantic formulation. In Section 3 we describe our proposal for the analysis of polar questions. Section 4 reviews and extends the account of Larsson et al. (2025) of wh-questions in SPA to incorporate the analysis we propose of polar questions. In Section 5 we report the results we obtained of a simulation of our account within the neuro-simulator Nengo (Bekolay et al., 2014). In Section 6, we conclude and describe future work.

[2]This finding is based both on a corpus study (in the Providence Corpus (American English) (**?**) showing that children provide answers to certain wh–questions **before** 'yes'/'no' are produced. It is also based on experimental work involving elicitation studies (in German and Chinese).

2 Background

2.1 The semantics of polar questions

There are a wide range of approaches to the semantics of questions (see Wiśniewski, 2015 for a review.). Nonetheless, on just about all existing theories, the entities denoted by wh-interrogatives are more complex entities than those denoted by polar-interrogatives. For instance, in what is probably the most commonly held view, the approach of (Hamblin, 1973), polar interrogatives denote the set $\{p, \neg p\}$, whereas wh-questions denote a potentially large set of possible candidate entities that might instantiate the property in question; in the partition theory (Groenendijk and Stokhof, 1997), a polar-interrogative denotes a partition that has exactly two entities (corresponding to the question being resolved positively or negatively, respectively.) With a unary wh-interrogative, the size of the partition is less determinate, it depends on the number of possible candidate entities that could serve as fillers for the property in question—if that number is known to be n, then the size is 2^n.

Related to this is the fact that the notion of answerhood provided by such theories recognizes for a polar question p? two answers, p and its negation $\neg p$. The motivation for this is that when interrogatives are embedded by *resolutive* predicates (e.g., 'know' or 'discover') they coerce the interrogative to denote a proposition that resolves the question (Lahiri, 2002; Ginzburg and Sag, 2000) and for polar questions these are p and $\neg p$. As pointed out by (Ginzburg, 1995; Omar, 2020) there are a variety of other propositions that count as *about* a question, without potentially resolving it:

(2) a. Jill: Is Millie leaving tomorrow? Bill: Possibly/It's unlikely/If it doesn't rain.

b. Bill provided information about whether Millie is leaving tomorrow. ((Ginzburg, 1995), ex., (99,101))

Corpus studies indicate that such answers are not rare: in a study of polar answers in (a subcorpus of) the Switchboard corpus by (de Marneffe et al., 2009) non-polar answers constitute 21/638 responses, whereas in a study of indirect answers to

polar questions building subcorpora from Switchboard and the Circa corpora (Wang et al., 2024) provide figures of 55% (Movie domain), 7% (Travel domain), and 34.7% (Tennis domain) (n=900) for responses not entailing 'yes' or 'no' (though these presumably also include cases classified as 'evasive responses' (e.g., 'I don't know', 'Not a question I would like to answer etc) in the taxonomy of responses to questions of (Ginzburg et al., 2022).).

2.2 SPA (Semantic Pointer Architecture)

SPA utilizes *holographic reduced representations* (HRR; Plate, 2003). These are a particular implementation of compressed representations, that is, (higher-order) semantic representations that are obtained by "compressing" (lower-level) semantic representations (Hinton, 1990). HRRs achieve this with two key operations: vector addition that simulates conjunction, and *circular convolution*, a multiplication operation that binds high-dimensional vectors of dimension d into a new vector of dimension d. HRR is a true-to-dimension instance of a *Vector Symbolic Architecture* (VSA; Gayler, 2003), in contrast to, for instance, tensor products (Smolensky, 1990). This feature of HRR is crucially for biological plausibility in terms of space utilized for representation (Stewart and Eliasmith, 2012)

The vector algebra of HRR includes the following operators ($\mathbf{a}$, $\mathbf{b}$, ... are (normalized unit) vectors.):

- $+$: $\mathbf{a}+\mathbf{b} = [\mathbf{a}_0+\mathbf{b}_0, \mathbf{a}_1+\mathbf{b}_1, \ldots, \mathbf{a}_{d-1}+\mathbf{b}_{d-1}]$
- $-$: $\mathbf{a}-\mathbf{b} = [\mathbf{a}_0-\mathbf{b}_0, \mathbf{a}_1-\mathbf{b}_1, \ldots, \mathbf{a}_{d-1}-\mathbf{b}_{d-1}]$
- $\circledast$: $\mathbf{c} = \mathbf{a} \circledast \mathbf{b} : \mathbf{c}_j = \sum_{k=0}^{d-1} \mathbf{a}_k \mathbf{b}_{j-k \pmod d}$
- inverse: $\mathbf{a}' = [\mathbf{a}_0, \mathbf{a}_{d-1}, \mathbf{a}_{d-2}, \mathbf{a}_{d-3}, \ldots, \mathbf{a}_1]$

$+$ and $\circledast$ have many properties in common with both scalar and matrix multiplication. They are commutative, associative, and satisfy distributivity. In this approach, contents for NL expressions can be compositionally built up with potential recursion using these operations *on vectors*, as exemplified in (3); (here and below *dog*_52,*cat*_63, etc. are vectors corresponding to a fixed entity, or class thereof.)

(3) a. The dog chases the cat $\mapsto$ **chase**$\circledast$**rel**$+$**dog_52**$\circledast$**agent**$+$**cat_63**$\circledast$**theme**

b. John thinks the dog chases the cat $\mapsto$ **John_17** $\circledast$ **theme** $+$ **think** $\circledast$ **rel** $+$ (**chase**$\circledast$**rel**$+$**dog_52**$\circledast$**agent**$+$**cat_63**$\circledast$**theme**) $\circledast$ **proparg**

The fact that any vector **a** has an approximate inverse **a'** (for which **a** $\circledast$ **a'** ≈ 1) plays a key role in enabling a theory of questions to be developed. Given a structure such as

(4) **agent** $\circledast$ **dog**$_{52}$ $+$ **frame** $\circledast$ **chase** $+$ **theme** $\circledast$ **cat**$_{43}$

"Who does dog$_{52}$ chase?" amounts to unbinding (4) with **theme**$'$ and thereby retrieving an answer.

(5) (4) $\circledast$ **theme**$'$ $=$ **agent** $\circledast$ **dog**$_{52}$ $\circledast$ **theme**$'$ $+$ **frame** $\circledast$ **chase** $\circledast$ **theme**$'$ $+$ **theme** $\circledast$ **cat**$_{43}$ $\circledast$ **theme**$'$ = noise + **cat**$_{43}$ $\approx$ **cat**$_{43}$

It is noteworthy that the true-to-dimension HRR vector manipulations are lossy: in particular the inverse $\mathbf{a}'$ of a vector **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).

2.3 Type Theory with Records (TTR)

Our general strategy to semantic analysis is to formulate analyses in a high level (symbolic) semantic framework, here Type Theory with Records (TTR). (Cooper, 2023; Chatzikyriakidis et al., 2025). An important motivation for using TTR is that all complex TTR objects are constructed from labelled sets, which correspond to the representation of structured objects which SPA achieves using superposition and circular convolution. This is the basis for a systematic mapping to SPA (Larsson et al., 2023), which in turn has a direct mapping into neural networks via the *Neural Engineering Framework* (NEF; Eliasmith and Anderson, 2003) which implements vectorial representations and manipulations in neural simulations.[3] How many levels of explanation are necessary is an open question, but as Eliasmith and Kolbeck (2015, e.g., p. 331) demonstrate, it is often necessary to '[be] thinking about multiple levels simultaneously when constructing information-processing systems.'

[3] See https://www.nengo.ai/.

We sketch here those aspects of TTR that are used in this paper. $s : T$ 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 'woman(a)', 'see(a,b)' where a, b : *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 : woman(a) can be glossed as "s is a situation which shows (or proves) that a is a woman".

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 (6(i)) represents the type of records which can be used to model situations where a woman runs, whereas (6(ii)) represents a record of that type where a : *Ind*, s : woman(a) and e : run(a):

(6) (i)
$$\begin{bmatrix} \text{ref} & : & \textit{Ind} \\ \text{c}_{\text{woman}} & : & \text{woman(ref)} \\ \text{c}_{\text{run}} & : & \text{run(ref)} \end{bmatrix}$$

(ii)
$$\begin{bmatrix} \text{ref} & = & a \\ \text{c}_{\text{woman}} & = & s \\ \text{c}_{\text{run}} & = & e \\ \dots & & \end{bmatrix}$$

3 Questions: Semantic analysis

We will assume a view of questions as propositional functions, a view apparently initiated by Ajdukiewicz (1926), developed significantly in Kubinski (1960), and subsequently shared and further developed by a number of different approaches, e.g., Krifka (2001). This view offers (a) a direct explanation of short answers to questions (Jacobson, 2016) —the short answer corresponds to the 'missing argument' and (b) is the basis for the most detailed account of answerhood (Ginzburg and Sag, 2000, pp. 129–149) that we are aware of.

The simplest way to implement a propositional abstract in TTR, which we adopt here, is to treat a question as being an entity as in (7a), that is, a function from objects of a particular type returning a type that can be used as a proposition. A 'who'–question is exemplified in (7b), a 'where'–question is exemplified in (7c), and a multiple wh-question is exemplified in (7d):

(7) a. $\lambda x{:}T_1 \,.\, T_2$ where T_1 and T_2 are types.

b. Who chased the cat $\mapsto$
$\lambda x : \textit{Ind} \,.\, \left[\text{c1} : \text{Chase}(x,\text{c})\right]$

c. Where is the cat $\mapsto$
$\lambda l : \textit{Loc} \,.\, \left[\text{c2} : \text{In}(l,\text{c})\right]$

d. Who chased whom $\mapsto$
$\lambda x : \textit{Ind}, y : \textit{Ind} \,.\, \left[\text{c1} : \text{Chase}(x, y)\right]$

In past work on questions in TTR questions were treated as 0-ary abstracts, implemented as abstraction over an empty set or as unrestricted abstraction. Here we propose to unify polar-questions with wh-questions and identify the 'missing element' for polar questions as a propositional modifier, drawing on the data in (2). This yields a content exemplified in (8).

(8) Did the dog chase the cat $\mapsto$
$\lambda P : \textit{PropMod} \,.\, \left[\text{c3} : P(\text{Chase(d,c)})\right]$

As a consequence, the space of simple answers to the question—the instantiations of the propositional abstract the question is assumed to denote– is broadened widely for a wide range of answers such as *probably*, *certainly*, *unlikely*, *certainly not*, and even conditional answers such as *if it does not rain* can be included. This also yields a particular simple view of what 'yes' and 'no' denote as short answers, which gives rise to a unified account

together with answers to wh–questions: 'Yes' on this view is the identity function, whereas 'No' is a function that is the negation function for positive polar questions and the identity for negative polar questions, given the data in (9):

(9) a. Yes $\mapsto \lambda p : Type.p$

b. No $\mapsto$ $\{\lambda p : Type.\neg p \;\; q$: positive pol-question
$\lambda p : Type.p \;\; q$: negative pol-question$\}$

c. A: Do you want coffee? B: No (I don't want coffee).

d. A: You don't want coffee? B: No (I don't want coffee).

On this view polar questions involve searching for propositional modifiers in contrast to 'who'-questions and 'where'-questions, which which involves searches for (human) individuals and locations, respectively. This offers the required underpinning for the developmental finding by (Moradlou et al., 2021, p. 175) discussed in section 1: '*wh-questions emerge earlier because the answer that can be provided to such questions in "training sessions" between carer and child is easier to ground perceptually than the abstract entities expressed by propositional answers required for polar-questions.*'

4 Integration of semantic analysis of questions and neural modeling

4.1 Question construction in SPA

This paper builds directly on the framework developed by Larsson et al. (2025), which proposes a hybrid semantic–neural account of question answering, integrating a type-theoretic analysis of questions with the Semantic Pointer Architecture (SPA).

A λ-structure is encoded in SPA as a structured semantic pointer comprising (i) a vector representing the ptype corresponding to the question body, (ii) a λ-path pointer indicating the role or field in the ptype where the abstracted value is to be inserted, and (iii) a type constraint on the abstracted variable:

(10) a. $\lambda v : Ind.\text{chase}(\text{dog}_{52}, v)$

b. $\mathbf{Q} = \begin{pmatrix} \textbf{lambdapath} \circledast \textbf{arg2}+ \\ \textbf{lambdatype} \circledast \textbf{Ind}+ \\ \textbf{body} \circledast \begin{pmatrix} \textbf{pred} \circledast \textbf{chase}+ \\ \textbf{arg1} \circledast \textbf{dog}_{52}+ \\ \textbf{arg2} \circledast \mathbf{I} \end{pmatrix} \end{pmatrix}$

with $\mathbf{I}$ the identity vector $\mathbf{I} = [1, 0, 0, \ldots]$, such that for any vector $\mathbf{x}$, $\mathbf{I} \circledast \mathbf{x} = \mathbf{x}$. This is a variant of de Bruijn indexing (de Bruijn, 1972), coding lambda terms without using variables but using paths to mark positions instead.

Question answering is modelled as a function application and memory retrieval over such representations. λ-application is approximated by vector symbolic operations that replace the abstracted component of the question-body ptype with an answer vector, yielding a fully specified ptype:

(11) $f(Q, A) = Q \circledast \textbf{body}' - Path + Path \circledast A$
where $Path = Q \circledast \textbf{lambdapath}'$

For the specific $Q = \mathbf{Q}$ given above and $A = \mathbf{A}$ given in (12a), $Path = Q \circledast \textbf{lambdapath}'$ evaluates (with some noise) to **arg2**, hence yielding, as shown in (12b) the desired answer:

(12) a. $\mathbf{A} = \textbf{cat}_{43}$

b. $\mathbf{P} = \begin{pmatrix} \textbf{pred} \circledast \textbf{chase}+ \\ \textbf{arg1} \circledast \textbf{dog}_{52}+ \\ \textbf{arg2} \circledast \mathbf{I} \end{pmatrix} - \textbf{arg2} + \textbf{arg2} \circledast \textbf{cat}_{43} \approx \begin{pmatrix} \textbf{pred} \circledast \textbf{chase}+ \\ \textbf{arg1} \circledast \textbf{dog}_{52}+ \\ \textbf{arg2} \circledast \textbf{cat}_{43} \end{pmatrix}$

Memory retrieval exploits the structural similarity between the question-body ptype and stored ptypes in associative memory, where candidate ptypes are retrieved via unbinding and clean-up operations. Lastly, the abstracted value is extracted by unbinding with the λ-path pointer. Type compatibility between questions, answers, and ptypes is enforced using vector-encoded type information and clean-up over a domain of types.

In extending the SPA–TTR framework from wh-questions to polar questions, the modifier is represented as a separate meta-propositional field. The new proposition is augmented with a modifier slot, and the original proposition itself becomes an argument.

Accordingly, a polar question abstracts over the modifier field inside the nested structure rather than over an unstructured response token. A *Yes* answer

corresponds to binding the modifier role to a value YES, while *No* corresponds to binding it to a negative value NO. This strategy cleanly separates propositional content from modification, supports short answers, and avoids destructive vector negation.

(13)

$$\mathbf{Q} = \begin{pmatrix} \mathbf{lambdapath} \circledast \mathbf{modifier}+ \\ \mathbf{lambdatype} \circledast \mathbf{modifiertype}+ \\ \mathbf{body} \circledast \begin{pmatrix} \mathbf{modifier} \circledast \mathbf{I}+ \\ \mathbf{arg1} \circledast \begin{pmatrix} \mathbf{pred} \circledast \mathbf{chase}+ \\ \mathbf{arg2} \circledast \mathbf{dog}+ \\ \mathbf{arg3} \circledast \mathbf{cat} \end{pmatrix} \end{pmatrix} \end{pmatrix}$$

For example, with an answer such as (14), we expect to obtain a ptype such as (15) by inputting **Q** and **A** into a SPA network for λ-application.

(14) $\mathbf{A} = \mathbf{Yes}$

(15) $$\mathbf{P} = \begin{pmatrix} \mathbf{modifier} \circledast \mathbf{Yes}+ \\ \mathbf{arg1} \circledast \begin{pmatrix} \mathbf{pred} \circledast \mathbf{chase}+ \\ \mathbf{arg2} \circledast \mathbf{dog}+ \\ \mathbf{arg3} \circledast \mathbf{cat} \end{pmatrix} \end{pmatrix}$$

Using the schema in (11), the computation above outputs the body of the question with the λ- abstracted variable replaced by the argument, yielding the form indicated in (15).

4.2 Answer extraction from a polar question ptype

So far, we have focused on how to construct a complete propositional type by combining the given question and answer. In naturalistic question answering, however, the more common situation is the converse: a question is given as input to the network, and an appropriate answer must be extracted from a proposition stored in memory a priori. In this setting, the network is assumed to have access to a ptype **P** encoding a proposition that resolves the question **Q**, and the task is to retrieve the corresponding answer value **A**.

In Larsson et al. (2025), a type checking mechanism is implemented via a reconstruction test: after a candidate proposition is retrieved from memory, the model attempts to recombine it with the question's $\lambda-$abstraction, and type compatibility is determined by whether the resulting vector can be successfully cleaned up to a proper ptype. Failure is taken to indicate a type mismatch. The present work adopted this mechanism and extended it to polar questions by replacing abstraction over individuals with abstraction over **modifier** values.

The answer-extraction process utilizes the same structural information encoded in the question representation that is used for λ application. In particular, the λ-path field of the question specifies the role or position within the ptype from which the answer is to be retrieved. Answer extraction can therefore be implemented by unbinding the stored ptype with the inverse of the λ-path pointer supplied by the question:

(16) $f_{qp}(Q, P) = P \circledast (Q \circledast \mathbf{lambdapath}')'$

Intuitively, the modifier is first unbound with **lambdapath** to recover the role corresponding to the abstracted argument. This role pointer is then inverted and used to unbind the stored ptype, yielding the filler associated with that role.

For the example discussed above, where the question is *Does dog chase cat?* and the stored ptype encodes the proposition (15), the computation proceeds as:

(17) $\mathbf{A} = f_{qp}(\mathbf{Q}, \mathbf{P}) = \mathbf{P} \circledast (\mathbf{Q} \circledast \mathbf{lambdapath}')' \approx \mathbf{P} \circledast (\mathbf{modifier})' \approx \mathbf{Yes}$

5 A proof-of-concept model

5.1 Neural simulation results

To see if λ-abstracted question answering is feasible with neural network models, we implemented the key steps of Section 4 in Nengo (https://www.nengo.ai/).The encoded proposition reservoir, from which an answer is to be found, consists of 5 propositions, represented by role-filler semantic pointers (vectors of 256 or 512 dimensions). The contents of the propositions are the following: P1 = *dog chases cat*, P2 = *cat does not like mouse*, P3 = *mouse sees dog*, P4 = *cow does not chase cat*, P5 = *dog likes cow*. The inputs to the network are three questions, posed one after the other: 1. *Does dog chase cat?* 2. *Does cat like mouse?* 3. *Does mouse see dog?* Answers corresponding to the questions can be found in propositions 1, 2, and 3, respectively. To obtain the answers from the

propositions, the networks unbind the body and the λ-path from the question, subtract the λ-path, and unbind both to retrieve the value bound with the propositional modifier (i.e., compare the resulting vector to the semantic pointers in clean-up memory).

In (Larsson et al., 2025), an intermediate memory clean-up step, which, by design, is to use an index of ptypes for searching the answer, was introduced to reduce noise. In its implementation, an auto-associative module was used to achieve the desired outcomes. This module is preserved. After the ptypes **P1** to **P5** unbind the body from **Q** (see "Memory Input"), the intermediate output semantic pointer is processed by an auto-associative memory clean-up to pin point the most similar ptype by a *winner takes all* mechanism (see "Memory Output").

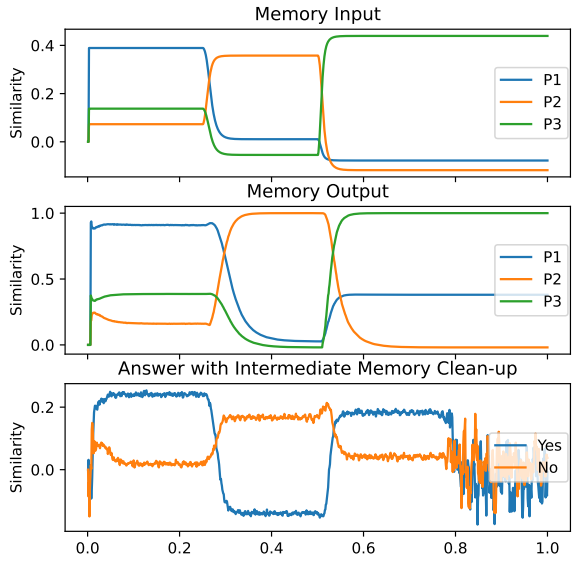


Figure 1: Retrieving a fragment answer according to 4.2 from a neural simulation. The neural simulation is dynamic because it runs in real time. The elapsed time (in seconds) is shown on the x-axis. The input to the network changes over time: from 0 to 0.25 the input is a semantic pointer corresponding to the question *Does dog chase cat?*, from 0.25 to 0.5 to the question *Does cat like mouse?*, and from 0.5 to 0.75 to *Does mouse see dog?*. From 0.75 onwards, no question is asked.

The simulation result is shown in the bottom row in Figure 1 ("Answer with Intermediate Memory Clean-up") and indicates the network's capacity to retrieve answers from encoded propositions via the λ-calculus. As is clear from the intermediate Figure 1 (which shows that winning P has similarity near 1 in all cases), the robustness issues arising from noise, exemplified in the bottom Figure 1 are not in relation to finding the right ptype but only in extracting out the polarity from the ptype once it has been identified. Thus, if the simulation instead looked solely for non-elliptical answers (answering with the whole ptype) we obtain near perfect performance. Robustness issues are likely due to unbinding from **P** with a noisy path ($\mathbf{Q} \circledast \mathbf{lambdapath}'$).

5.2 Robustness test results

Since representations and operations in the SPA rely on randomly initialized high-dimensional vectors, the behavior of the model is, in principle, sensitive to the choice of the initial state, i.e., a random seed. To assess the robustness of the proposed question-answer mechanism, a systematic evaluation was conducted. Specifically, we sampled 100 random seeds. For each seed, the full question-answering procedure was executed on the same set of 3 test questions. We used the number of correctly answered questions per trial, a score between 0 and 3, as the metric. 2 conditions were evaluated: 256- and 512-dimensions. As shown in Figure 2, performance is more stable at higher dimensionality: the 256-dimensional condition exhibits greater variability across random initializations, whereas the 512-dimensional condition shows a stronger concentration of perfect scores. This indicates that the proposed question-answering mechanism is robust to random initialization, more consistent at higher dimensionalities, and that human-like performance is not driven by idiosyncratic properties of specific random seeds.

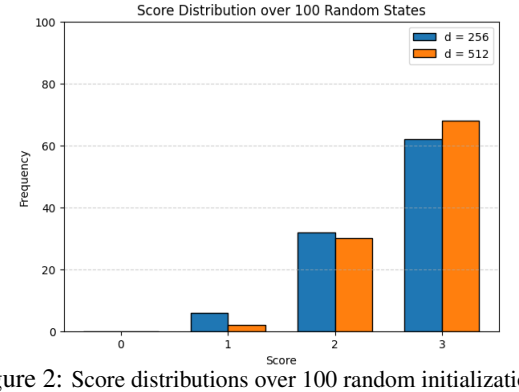


Figure 2: Score distributions over 100 random initializations for the polar question task. Each histogram shows the fraction of the three test questions answered correctly per trial for semantic pointer dimensionalities of 256 (blue) and 512 (orange). Higher dimensionality yields more consistent performance.

6 Concluding Remarks and Future Work

In this paper we offer an extension of an earlier proposal for treating wh-questions within a compositional, neurally–implemented semantic framework

that interfaces with memory to the other main class of questions, namely polar questions. Our proposal yields improved empirical coverage for polar questions as compared with previous formal semantic accounts. It also offers the basis for an account of the finding by (Moradlou et al., 2021) that understanding for wh-questions emerges in language development *before* that of polar questions—a finding that goes against previous accounts of questions in which polar questions are simplest in terms of their semantic complexity.

However, our account is still preliminary in a number of important respects that require developments in neural semantics and neural modelling of memory.

Answer Selection As mentioned in Section 4, the **modifier** term can contain different answers ranging from short answers to more complex conditional answers. Implementing such an enriched answer space in SPA would requires mechanisms beyond the binding and clean-up operations employed at current stage. In particular, distinguishing between graded or conditional answers plausibly depends on relations of entailment or compatibility between propositions, rather than on simple structural matching. Though SPA supports approximate similarity-based reasoning, modeling answerhood in this richer sense requires integrating explicit logical constraints, graded inference (Sadrzadeh et al., 2018), and probabilistic representations (Furlong et al., 2022) into our account. This is also necessary for developing answer selection strategies akin to exhaustivity.

Memory compartmentalization in SPA A central issue highlighted by the present work concerns the treatment of memory in SPA. The SPA we currently employ represents memory as a collection of semantic pointers in its vocabulary, without a principled architectural distinction between different memory systems such as long-term memory and its various subsystems (episodic, semantic, procedural etc) and working memory. Despite recent efforts to bridge this gap (Gosmann and Eliasmith, 2021), memory in SPA is largely modelled as a uniform representational substrate. From a computational perspective, this lack of compartmentalization limits the transparency. In the current model, propositional representations that function as background knowledge and those that are actively manipulated during question answering are treated uniformly. However, the operations involved in answering a question naturally decompose into stages with distinct functional roles: propositions must first be stored in a relatively stable form, plausibly corresponding to long-term or short-term memory, while the receipt of a question initiates a transient sequence of transformations that are more naturally associated with working memory.

Besides, a more articulated memory architecture is also desirable on linguistic and cognitive grounds. Empirical work suggests that language comprehension involves dynamic interaction between stable knowledge representations and temporarily maintained structures that are sensitive to the current conversational context (Daneman and Merikle, 1996). Treating all memory as a single homogeneous store risks conflating these distinct roles, whereas a whole network with distinct mechanisms might better reflect current views in psycholinguistics and cognitive neuroscience.

Acknowledgments

We thank three Brigap reviewers for their useful comments. This paper received support from Université Paris Cité as part of its program "Initiative d'excellence" IdEx (ANR-18-IDEX-0001) and from ANR CODIM.

References

Kazimierz Ajdukiewicz. 1926. Analiza semantyczna zdania pytajnego. *Ruch Filozoficzny*, 10:194b–195b.

Trevor Bekolay, James Bergstra, Eric Hunsberger, Travis DeWolf, Terrence C Stewart, Daniel Rasmussen, Xuan Choo, Aaron Voelker, and Chris Eliasmith. 2014. Nengo: a python tool for building large-scale functional brain models. *Frontiers in Neuroinformatics*, 7:48.

Galina B Bolden, John Heritage, and Marja-Leena Sorjonen. 2023. Chapter 1. introduction: Polar questions and their responses. *Responding to polar questions across languages and contexts*, pages 1–39.

Stergios Chatzikyriakidis, Robin Cooper, Eleni Gregoromichelaki, and Peter Sutton. 2025. *Types and the Structure of Meaning: Issues in Compositional and Lexical Semantics*. Cambridge Elements in Semantics. Cambridge University Press.

Robin Cooper. 2023. *From Perception to Communication: a Theory of Types for Action and Meaning*. Oxford University Press.

Meredyth Daneman and Philip M. Merikle. 1996. Working memory and language comprehension: A meta-analysis. *Psychonomic Bulletin & Review*, 3(4):422–433.

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.

Marie-Catherine de Marneffe, Scott Grimm, and Christopher Potts. 2009. Not a simple yes or no: Uncertainty in indirect answers. In *Proceedings of the SIGDIAL 2009 Conference*, pages 136–143.

Chris Eliasmith. 2013. *How to build a brain: A neural architecture for biological cognition*. OUP USA.

Chris Eliasmith and Charles H. Anderson. 2003. *Neural Engineering*. Computational Neuroscience. MIT Press, Cambridge, MA.

Chris Eliasmith and Carter Kolbeck. 2015. Marr's attacks: On reductionism and vagueness. *Topics in cognitive science*, 7(2):323–335.

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.

Jerry A Fodor. 1974. Special sciences (or: The disunity of science as a working hypothesis). *Synthese*, pages 97–115.

Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Gottlob Frege. 1918. Der Gedanke. *Beiträge zur Philosophie des deutschen Idealismus*, 1(2):58–77.

P Michael Furlong, Terrence C Stewart, and Chris Eliasmith. 2022. Fractional binding in vector symbolic representations for efficient mutual information exploration. In *Proceedings of the ICRA Workshop: Towards Curious Robots: Modern Approaches for Intrinsically-Motivated Intelligent Behavior*, pages 1–5.

Ross W. Gayler. 2003. Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience. *ICCS/ASCS International Conference on Cognitive Science*, pages 133–138. Slightly updated 2004 in paper cs/0412059 on arXiv.

Ross W Gayler. 2004. Vector symbolic architectures answer jackendoff's challenges for cognitive neuroscience. *arXiv preprint cs/0412059*.

Jonathan Ginzburg. 1995. Resolving questions, i. *Linguistics and Philosophy*, 18:459–527.

Jonathan Ginzburg and Ivan A. Sag. 2000. *Interrogative Investigations: the form, meaning and use of English Interrogatives*. Number 123 in CSLI Lecture Notes. CSLI Publications, Stanford: California.

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. 2021. Cue: A unified spiking neuron model of short-term and long-term memory. *Psychological Review*, 128(1):104.

Jeroen Groenendijk and Martin Stokhof. 1997. Questions. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Linguistics*. North Holland, Amsterdam.

C. L. Hamblin. 1973. Questions in montague english. In Barbara Partee, editor, *Montague Grammar*. Academic Press, New York.

Geoffrey E. Hinton. 1990. Mapping part-whole hierarchies into connectionist networks. *Artificial Intelligence*, 46(1):47–75.

Pauline Jacobson. 2016. The short answer: implications for direct compositionality (and vice versa). *Language*, 92(2):331–375.

Manfred Krifka. 2001. For a structured meaning account of questions and answers. *Audiatur Vox Sapientia. A Festschrift for Arnim von Stechow*, 52:287–319.

Tadeusz Kubinski. 1960. An essay in the logic of questions. In *Proceedings of the XIIth International Congress of Philosophy (Venetia 1958)*, volume 5, pages 315–322, Firenze. La Nuova Italia Editrice.

Utpal Lahiri. 2002. *Questions and Answers in Embedded Contexts*. Oxford University Press, Oxford.

Staffan Larsson, Robin Cooper, Jonathan Ginzburg, and Andy Luecking. 2023. Ttr at the spa: Relating type-theoretical semantics to neural semantic pointers. In *Proceedings of the 4th Natural Logic Meets Machine Learning Workshop*, pages 41–50.

Staffan Larsson, Jonathan Ginzburg, Robin Cooper, and Andy Lücking. 2025. Finding answers to questions: Bridging between type-based and computational neuroscience approaches. In *16th International Conference on Computational Semantics*, page 128.

Sara Moradlou, Xiaobei Zheng, TIAN Ye, and Jonathan Ginzburg. 2021. Wh-questions are understood before polar-questions: Evidence from english, german, and chinese. *Journal of Child Language*, 48(1):157–183.

AGHA Omar. 2020. Non-resolving responses to polar questions: A revision to the qud theory of relevance1. *Proceedings of Sinn und Bedeutung 24 Volume*.

Tony Plate. 2003. *Holographic Reduced Representation: distributed representation for cognitive structures*.

Mehrnoosh Sadrzadeh, Dimitri Kartsaklis, and Esma Balkır. 2018. Sentence entailment in compositional distributional semantics. *Annals of Mathematics and Artificial Intelligence*, 82(4):189–218.

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 Stewart and Chris Eliasmith. 2012. Compositionality and biologically plausible models. In M. Werning, W. Hinzen, and E. Machery, editors, *The Oxford handbook of compositionality*, pages 596–615. Oxford University Press.

Aaron R Voelker and Chris Eliasmith. 2023. Programming with semantic pointers: A practical guide. *Journal of Cognitive Neuroscience*.

Zijie Wang, Farzana Rashid, and Eduardo Blanco. 2024. Interpreting answers to yes-no questions in dialogues from multiple domains. *arXiv preprint arXiv:2404.16262*.

Andrzej Wiśniewski. 2015. Semantics of questions. In *Handbook of Contemporary Semantic Theory, second edition*, Oxford. Blackwell.

Artificial Language Learning Paradigm Reveals Pragmatic Blind Spots in Vision-Language Models

Yan Cong[1], **Julia Rayz**[2]
[1]School of Languages and Cultures, Purdue University
[2]School of Applied and Creative Computing, Purdue University
{cong4, jtaylor1}@purdue.edu

Abstract

Humans are pragmatic language users who naturally and effortlessly reason about the choice of utterances that help collaborate and engage in social interactions. In this paper, we examine whether vision-language models (VLMs) exhibit similar pragmatic reasoning effects through a validated artificial language learning paradigm. Across four experiments, we evaluate five VLMs' sensitivity to production cost, ambiguity-driven competition effects, and the influences of visual features and model properties. We find evidence of cost effects in some VLMs. However, no model consistently exhibits competition effects driven by ambiguity risk, a hallmark of Gricean pragmatic reasoning. We also find that model scale alone does not predict pragmatic alignment; architectural choices play a larger role. Moreover, probability-based methods reveal clearer effects than prompting. Overall, current VLMs capture only a restricted subset of pragmatic effects central to Gricean reasoning, suggesting gaps in multimodal pragmatic reasoning.

1 Introduction

In humans' daily conversations, utterances and their *alternatives* (expressions that have the same literal semantics and that a speaker could have used but chose not to) compete. Rational speakers choose expressions that are true, informative, relevant, and unambiguous, while listeners infer the speaker's intended meaning from those choices (Grice, 1989; Geurts, 2010). For example, the speech act of using ambiguous expressions violates Grice's *Manner* Maxim (avoid ambiguity) and leads the listener to draw a manner implicature inference: the speaker *intends* to convey a non-literal, *disambiguation*-based inference (Grice, 1989; Rett, 2015, 2019). This speaker-listener iterative process of pragmatic reasoning rests on evaluating the competition of *alternatives*. Recent Gricean pragmatics theories recast the notions of cost, including utterance length, syntactic and phonological complexity, or dependency structure, as refined dimensions shaping speakers' reasoning (Katzir, 2007; Buccola et al., 2021).

Our study concerns vision-language models (VLMs) as pragmatic language users. We build on the Artificial Language Learning (ALL) framework studied in Buccola et al. (2018), which shows that humans engage in pragmatic reasoning when even using nonce expressions in minimal communicative systems, suggesting that humans instinctively and spontaneously apply strategic reasoning. ALL provides a controlled way to study language use by isolating specific inferential strategies and pressures such as cost and competition. We adopt this paradigm to examine whether similar pragmatic reasoning patterns emerge in VLMs, exploring whether VLMs display sensitivity to the same pressures and strategies that influence human pragmatic reasoning. Evidence of such emergence can shed light on the development and optimization of pragmatically aware VLMs (Ma et al., 2025).

Our results align with and add to previous research by contributing three diagnostic aspects in VLM assessment: (a) scale invariance about model size (7B to 13B); (b) the generation-representation gap (probability vs. prompting); and (c) architectural factors (e.g., Janus showed cost effects that other models lack).[1]

2 Background

2.1 Pragmatic reasoning in humans

Humans are pragmatic language users who reason about alternatives to derive enriched, implied meaning. Under Gricean probabilistic frameworks (Grice, 1989, 1975; Bergen et al., 2016; Franke and Bergen, 2020; Frank and Goodman, 2012; Franke and Jager, 2016; Qing and Frank, 2014; Franke,

[1]Scripts are available in this public repository: https://doi.org/10.17605/OSF.IO/E6TKM.

Third Workshop on the Bridges and Gaps between Formal and Computational Linguistics, pages 50–62

2011), listeners consider not only what a speaker says but also what they could have said yet chose not to, drawing inferences derived from competition with alternatives. Although most work focuses on alternatives and competition, the role of cost as an independent dimension in pragmatic inference remains relatively underexplored. Moreover, the intersection of cost and ambiguity-based (Manner) inference is less studied due to the relative lack of robustness, making it difficult to empirically study ambiguity-based implicature in humans (Wilson, 2017) and even fewer studies are reported in language models (Cong, 2024). We hope to bridge the gap and pragmatically capture ambiguity in VLMs.

2.2 The artificial language learning paradigm

Artificial Language Learning (ALL) paradigms are used to investigate linguistic universals and domain-general cognitive preferences by isolating minimal, explicitly defined alternatives (Martin et al., 2019; Buccola et al., 2018; Culbertson, 2012; Culbertson and Schuler, 2019; Motamedi et al., 2019). Using a minimal artificial lexicon, Buccola et al. (2018) show that human learners exhibit scalar-implicature-like behavior even in the absence of prior natural language experience, demonstrating that pragmatic competition can arise spontaneously from general reasoning dynamics, going beyond language-specific conventions. Cong (2021) used ALL to study *cost* and found that humans behave pragmatically but not optimally, showing a trend to use short, ambiguous expressions when production cost is high, and significantly adjusting their choices to ambiguity risk.

This line of work suggests that humans treat cost as a principled factor in pragmatic inference, which is manifested in (i) their preference to ambiguous phrases when the competing unambiguous alternative is costly to produce or when the competition is weak and the ambiguous phrase becomes effectively unambiguous, and (ii) the disappearance of such preference when both phrases have approximately equal cost. Overall, because of ALL's minimal ingredients and exclusion of prior linguistic knowledge, it is especially useful for us to diagnose whether VLMs exhibit emergent pragmatic behavior rather than reproducing surface patterns from training data.

2.3 Pragmatic capacity of VLMs

Previous studies shed light on referring expression generation and comprehension, visual encoders, and alignment of visual and textual representations (Shu et al., 2025; Ma et al., 2025; Fu et al., 2025; Cao et al., 2020; Salin et al., 2022; Shirnin et al., 2024; Liu et al., 2021; Hua and Artzi, 2024; Vaduguru et al., 2025). Pertaining to our work, Pitta et al. (2025) evaluated VLMs in relation to the visual entailment task, where models infer whether a given image supports a specific textual caption. They note that although some VLMs perform well on visual relation detection and linguistic alignment, they still show biases rooted in text processing. Kouwenhoven et al. (2024) used ALL to examine how VLMs can develop structured communication protocols, a crucial aspect for enhancing effective collaboration among agents in multi-agent systems (Lazaridou et al., 2018). No studies have been done to apply a theory-informed ALL paradigm to identify Gricean reasoning effects in VLMs, a critical step toward rigorous multimodal pragmatics benchmark.

3 Rationale and hypothesis

Inspired by Bergen et al. (2016); Buccola et al. (2018); Cong (2021), we hypothesize that a rational agent reasons about ambiguity-based inference by weighing the communicative benefits of disambiguation against the production costs of longer expressions. In any context where both a short but ambiguous form ϕ and a longer, more costly but unambiguous alternative ψ are logically true, the rational choice depends on a context-sensitive trade-off between the cost of producing ψ and the risk of miscommunication triggered by using ϕ. This leads to four specific predictions pinpointing pragmatic reasoning effects in VLMs:

1. `Cost effect`: When production costs are high (a long expression denotes the referent), a rational agent is predicted to choose the short, ambiguous form ϕ more often than when production costs are low.
2. `Competition effect`: When ambiguity risks are high (strong competition and no contextually salient reading), a rational agent is predicted to use the short, ambiguous form ϕ less often than when ambiguity risks are low.
3. `Lack of Effect`: Image features (color intensity and objects position) should not modulate pragmatic reasoning: VLMs responses should remain stable regardless of image features.
4. `Model Effect`: Depending on model's properties (scale and language backbone), the sign

and magnitude of the above effects are predicted to differ across VLMs.

We designed four experiments under the ALL paradigm to operationalize the measurement of the above pragmatic reasoning effects in VLMs. First, we examine the cost and competition effects by comparing VLMs to humans (as rational agents). Second, we examine the cost and competition effects by comparing VLMs to the *ideal* rational agents, ones that consistently favor the short, ambiguous phrase whenever it can truthfully denote the intended referent, always relying on listeners to reason about the speaker's choice and recover the intended meaning (Bergen et al., 2016; Franke and Bergen, 2020). Third, we examine how image features such as color intensity and object position influence VLMs' choices. Fourth, we examine how model properties such as parameter sizes and architectural choices influence VLMs' responses.

4 Experiments

4.1 Experiments for humans

The original ALL experiment in Buccola et al. (2018) and Cong (2021), on which our VLM experiments design is based, examined how human speakers (n=58) balance utterance cost and ambiguity when choosing among competing alternatives. Participants learn an artificial Martian language whose nonce phrases denote geometric objects. The lexicon includes three expression types: (i) short, ambiguous phrases that can refer to two objects, (ii) long (phonologically complex), unambiguous phrases, and (iii) longest, unambiguous phrases (the longest among the three types). The task is framed as a friendly interaction with Martians who value efficiency, preferring the shortest phrase that still ensures true, correct reference.

The experiment has a learning phase and a testing phase (Figure 1). In the learning phase, participants infer meanings from trial-by-trial feedback: each trial presents a nonce phrase as in Table 1 and three candidate objects, and participants guess the referent. Utterance cost is made salient by requiring more key-presses to reveal longest phrases.

In the testing phase, participants act as speakers: they view three objects horizontally in one display, one circled to denote that it is the intended target referent, and must type one of the learned phrases to communicate the target. No feedback is given, and instructions stress that Martians always choose the phrase that can concisely and uniquely identify the object, prompting participants to reason about cost and ambiguity. Participants complete both the *experimental* trials, where two geometric objects make the short phrase contextually ambiguous, and the *control* trials, where ambiguity risk is low, since only one geometric object is present and the other is replaced by a non-geometric distractor.

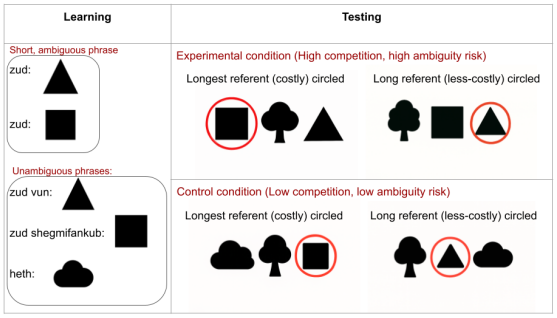


Figure 1: Artificial language learning design. Long(est) referent: a referent that the long(est) phrase denotes.

Referents	Phrases
Geometric shapes	**Short:** *kad, zud*; **Long:** *kad mof, zud mof, kad vun, zud vun*; **Longest:** *kad shegmifanzub, zud shegmifankub*
Organic shapes	*sot, heth*

Table 1: Nonce phrases used in our experiments.

We used the nonce phrases (Table 1) from Cong (2021), created following Buccola et al. (2021) and Buccola et al. (2018), which have phonetically and syntactically validated that these phrases do not exist in any natural languages. The design of these nonce phrases used sequence *length* as a proxy to *cost*. Controlled experiments in psychology show that the length of phrases signifies their conceptual complexity and cost, with longest phrases corresponding to more complex concepts (Pimentel et al., 2023; Lewis and Frank, 2016). Lewis and Frank (2016) suggested that human participants inherently link word length to conceptual complexity during both comprehension and production, even when factors like frequency, familiarity, imageability, and concreteness are considered.

4.2 Experiments for VLMs

Our VLM experiments are the same as the human ALL experiment in Buccola et al. (2018) and Cong (2021) (Figure 1), except for the following three differences, ensuring valid comparison and feasible translation of experiments between humans

and VLMs. First, we additionally examined image properties, including objects position and color intensity. As a consequence, we expanded the original dataset of 96 sets (unique combinations of image and phrase) into 768 sets. There were 36 base images, organized into 6 spatial configurations of three geometric objects (*triangle*, *square*, *circle*), in addition to 2 distractor organic objects (*cloud* and *tree*). Object positions were permuted to prevent spatial cues from biasing disambiguation. Each object was centered within its respective quadrant. Color intensity was manipulated using *solid* or *transparent* renderings. All the images were 1536x1024 pixels in size (objects did not fully occupy these pixels), with no text on the images. There were 16 unique phrase-object mappings. Each phrase denoted at least one object (Table 1).

Second, instead of learning via iterative correctness feedback, inspired by Verhoef et al. (2024); Pitta et al. (2025), we designed experiments in such a way that VLMs learn the mapping of phrase and object through the following structured declarative prompt: "*The label for this object is <label>*", pairing each nonce phrase with a visual reference in a single exposure per display, effectively a one-shot declarative learning setup. Namely, we give each shape in a single image (c.f., Figure 1 – Learning). If a phrase is ambiguous and can refer to two objects, it is learned in two separate displays.

We additionally conducted a supplementary, qualitative analysis on a single trial rather than the full dataset, examining the impact of prompting on model performance. Specifically, to address concerns that the models might lack sufficient information or context to exhibit pragmatic reasoning, a follow-up experiment was conducted using few-shot declarative learning together with an interactive Martian game context. For example, we showed three images where the object "zud" is a different color and in a different position each time. This setup explicitly provided the models with feature-invariance rules stating that color, size, and position are irrelevant to the object names, explicit cost instructions that shorter phrases are preferred if both candidate descriptions are true, and communicative framing that placed the model in a game with a listener in order to establish a shared conversational context. All the prompt templates were presented in the Appendix A.

Third, instead of typing phrases in the testing phase, we used two methods to probe VLMs. In forced-choice format, we prompted VLMs to generate and choose a phrase from their learned phrases. Model input included images of three objects and an instruction prompt in (1) `Default` plain form: "*Identify the label for the circled object in the image from the following options: <zud>, <zud vun>, <zud shegmifankub>, <heth>. Respond with only the label without explaining*"; and (2) in `Explicit` instruction form with the following instruction appended to the Default form "*Choose the option that concisely and uniquely identifies the circled object*" (Ma et al., 2025; Park et al., 2024). An Image was appended to the prompt accordingly (c.f., Figure 1 – Testing). VLMs output an option out of four. The ordering of the label was randomized at each trial to alleviate positional bias. The prompting variants (1,2) allowed us to examine if explicit efficiency instruction (as given to humans) would improve VLMs reasoning.

Besides prompting, we used forced-choice logprob scoring (Misra et al., 2020, 2021; Misra, 2022; Misra et al., 2023; Wolf et al., 2020) to compute phrase surprisals given an image and instruction prompts (as in the prompting method), normalized by phrase token length. Surprisal is the token-level negative log probability of the exact target token sequence under each VLM's native tokenizer, conditioned on the same multimodal input across VLMs. For each trial, we computed surprisal for all four phrases and took the highest-probability phrase as the model's choice, then compared these choices across models and conditions. We did not directly compare raw surprisal values across models or conditions, but instead derived a discrete choice from VLM surprisals first, to ensure valid comparisons.

We included five VLMs with differences in scale, visual encoding, and language backbone: Llava-1.5-7b and Llava-1.5-13b (Liu et al., 2023), Llava-1.6-7b (Liu et al., 2024), Gemma-2-3b (Steiner et al., 2024), and Janus-pro-1.3b (Wu et al., 2024).

4.3 Statistical tests

For Experiments 1 and 2 on cost and competition effects, to assess VLMs' alignment with human responses as reported in Cong (2021), we conducted chi-square tests of independence to determine whether the proportion of short, ambiguous responses differed significantly as a function of either *CircledObject* (longest≈costly vs. long≈less-costly; cost effect) or *Condition* (control vs. experimental; competition effect). To evaluate VLMs' alignment with ideal rational agents, we fit logistic

regression models in which match rates between each VLM's predicted phrases and the true labels were regressed on cost (*CircledObject*) and competition (*Condition*).

For Experiment 3, which examined the absence of image feature effects, we fit additional logistic regression models to test how color intensity (solid vs. transparent) affected these match rates. We applied Levene's tests to assess the homogeneity of variance in VLMs responses across object position. These tests evaluated whether VLMs response variability differed across trials with identical content and conditions but varied only in object spatial arrangement. p-values were adjusted for multiple comparisons (Benjamini–Hochberg false discovery rate (FDR)), with FDR-adjusted p-values below 0.05 indicating significant variance heterogeneity across images with different object positions.

In Experiment 4, we fitted linear mixed-effects models (LMEs) to investigate how different Llava model variants predict the accuracy of matches with an ideal rational agent. In these LMEs, match served as the dependent variable and model type as the fixed-effect predictor, and we specified a random intercept for the item identifier to account for the repeated-measures structure of the data. We constructed two LMEs: on parameter size by comparing Llava-1.5-13b with Llava-1.5-7b; on language model backbone by comparing Llava-1.6-7b with Llava-1.5-7b.

5 Results

5.1 Experiment 1: Cost and competition effects - VLMs vs. humans

5.1.1 Prompting-based metric

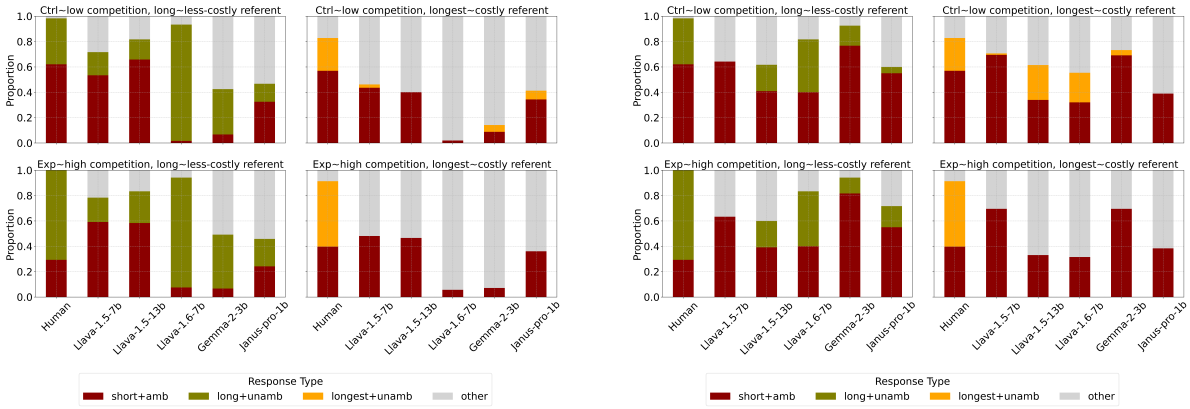
Figure 2a shows the proportion of each response type in VLMs' prompting-based responses, with "other" representing irrelevant or false responses, for instance, *zud vun* would be false in terms of literal semantics when the circled object can only be referred to using *zud* or *zud shegmifankub*. First, we observed a cost effect in Janus, which chose short, ambiguous expressions more often when cost was high (longest referent) than when cost was low (long referent), consistent with how humans perform in Cong (2021). Second, inconsistent with how humans perform in Cong (2021), we did not observe a competition effect in any of the VLMs: relative to control conditions with low ambiguity risk, the experimental conditions did not lead the models to choose short, ambiguous expressions less often. Third, relative to humans' responses in Cong (2021), we found a substantial proportion of "other" responses in Llava-1.6-7b and Gemma.

Model	Cost effect		Competition effect	
	OR	p	OR	p
Prompting-based response				
Llava-1.5-7b	1.52	0.009	1.22	0.193
Llava-1.5-13b	2.137	<0.0001	1.087	0.612
Llava-1.6-7b	1.22	0.748	3.552	0.004
Gemma-2-3b	0.827	0.632	0.851	0.652
Janus-1.3b	0.73	0.072	0.939	0.74
Probability-based response				
Llava-1.5-7b	0.772	0.134	0.993	1.000
Llava-1.5-13b	1.322	0.097	0.951	0.796
Llava-1.6-7b	1.429	0.033	0.984	0.977
Gemma-2-3b	1.682	0.006	1.097	0.622
Janus-1.3b	1.941	<0.0001	0.982	0.957
Humans	0.901	0.793	2.789	0.0002

Table 2: VLMs alignment with humans. OR (Odds Ratio) < 1 indicates lower odds of the short, ambiguous response when production costs are high (*cost effect*) or when ambiguity risks are low (*competition effect*), Odds Ratio > 1 indicates higher odds in such conditions; aligned (sign and magnitude); misaligned; improved with explicit instruction.

Statistical tests results were summarized in Table 2. Humans showed a strong competition effect but a weak cost effect (Cong, 2021): circling costly objects showed a small increase in short, ambiguous choices (48.28% for costly and 45.69% for less-costly), corresponding to just a 2.59% difference, and this pattern was not statistically significant ($\chi^2 = 0.069$, $p = 0.793$). Because humans themselves exhibit only a weak directional cost effect, we considered VLM as "aligned with humans" as long as it shows at least a 2.59% higher rate of short, ambiguous responses for costly relative to less-costly referents, matching the magnitude of the human pattern and ensuring that VLM-human comparisons reflect the empirical behavior of human participants.

Table 2 shows that statistically, relative to the behavioral threshold humans display, only Janus exhibit cost effect, and no VLMs show competition effect, suggesting that with the prompting method, VLMs struggle with choosing the true and relevant responses, especially in scenarios when a longest≈costly referent is circled (i.e., intended) and production cost is high (Figure 2a). We did not find solid evidence of competition effects, suggesting lack of sensitivity to ambiguity risks. With explicit instruction, only Llava-1.6-7b prompting showed cost effect (χ^2=6.054, p=0.013); and no VLMs showed competition effects (details in Ap-



(a) Prompting-based metric

(b) Probability-based metric

Figure 2: VLM responses in the testing phase, grouped by condition and circled object.

pendix Figure 4, 5).

As to the supplementary, qualitative experiment with more elaborate, interactive prompts, results suggested that all tested models (PaliGemma-2-3B, Janus-pro-1.3B, and Llava) failed to perform the task, and in many cases behaved worse than in the minimal one-shot setup. PaliGemma-2-3B repeatedly disregarded the negative constraint to respond with only the label and instead produced descriptive sentences listing the features it had been instructed to ignore, for example, statements such as *The object is square* and *The object is red*. In addition, the model frequently hallucinated objects and shapes that were not present in the dataset, including hearts, stars, and diamonds. Under the increased complexity introduced by the few-shot interactive prompt, Janus and Llava showed similar behaviors, with many "other" outputs.

5.1.2 Probability-based metric

Figure 2b shows proportions of each response type in VLMs' probability-based responses, with generally different patterns than the prompting responses. First, we observed cost effect in Llava-1.5-7b: when the intended referent is costly, it chose the short, ambiguous phrases more often than when the referent is less costly. Second, similar to prompting results in Figure 2a, we did not observe competition effects in any VLMs. Third, we observed generally fewer "other" in probability than in prompting responses. Statistically, Table 2 confirmed our observation that Llava-1.5-7b exhibited cost effect and that no VLMs aligned with humans in competition effects, suggesting that regardless of prompting or probability methods, VLMs did not show consistent sensitivity to ambiguity risks.

5.2 Experiment 2: Cost and competition effects - VLMs vs. ideal rational agent

5.2.1 Prompting-based metric

Figure 3a shows the percentage of match between VLM's prompting-based responses and true labels, which are always short, ambiguous phrases. Since there were four options in the testing phase, we drew the dotted line at 25% as the chance level match accuracy. First, we observed cost effect in Janus and Gemma, where there were more matches in "longest referent" (a scenario that the costly object is circled and intended) than in "long referent" (less-costly intended). Second, we observed no competition effects that more matches should be in control than in experimental condition, resonating with Experiment 1 (Figure 2a). Third, we observed that Llava-1.5-7b and Llava-1.5-13b consistently showed above chance match accuracies, across conditions and circled objects, suggesting high tolerance of ambiguity risks since short, ambiguous phrases were their frequent choices.

Table 3 shows that statistically, circling a long≈less-costly object decreased match odds for Gemma and Janus, confirming our observation about cost effects that when the intended is a costly referent, there is an increase in choosing short, ambiguous phrases. This also resonates with Experiment 1 where we found cost effect in Janus. Across VLMs, we did not find statistical evidence for competition effects, echoing Experiment 1. With explicit instruction, no VLMs showed cost effects; Llava-1.6-7b prompting showed competition effect (β=-0.7, p=0.024) (Appendix Figure 6, 7).

Model	Cost effect		Competition effect	
	β	p	β	p
Prompting-based response				
Llava-1.5-7b	0.429	0.013	0.191	0.189
Llava-1.5-13b	0.635	<0.0001	0.087	0.553
Llava-1.6-7b	0.0093	0.982	1.2675	0.004
Gemma-2-3b	-0.735	0.022	-0.178	0.523
Janus-1.3b	-0.441	0.017	-0.091	0.555
Probability-based response				
Llava-1.5-7b	-0.958	<0.0001	-0.019	0.907
Llava-1.5-13b	0.2569	0.145	-0.05	0.741
Llava-1.6-7b	0.588	0.001	-0.0118	0.939
Gemma-2-3b	0.331	0.100	0.093	0.569
Janus-1.3b	0.529	0.002	-0.019	0.898

Table 3: VLMs alignment with ideal rational agent. Values: β and p; aligned (sign and magnitude); misaligned; improved with explicit instruction.

Model	Default prompt		Explicit instruction	
	Acc	F1	Acc	F1
Prompting-based response				
Llava-1.5-7b	0.510	0.674	0.241	0.370
Llava-1.5-13b	0.527	0.685	0.241	0.370
Llava-1.6-7b	0.042	0.079	0.076	0.139
Gemma-2-3b	0.073	0.136	0.000	0.000
Janus-1.3b	0.318	0.480	0.290	0.448
Probability-based response				
Llava-1.5-7b	0.666	0.799	0.675	0.805
Llava-1.5-13b	0.368	0.537	0.291	0.447
Llava-1.6-7b	0.359	0.527	0.202	0.336
Gemma-2-3b	0.742	0.851	0.730	0.841
Janus-1.3b	0.468	0.634	0.425	0.593
Human			0.470	0.628

Table 4: Human and VLMs overall performance in matching with ideal rational agents. Acc: Accuracy.

5.2.2 Probability-based metric

Figure 3b demonstrates the percentage of match between VLM's probability-based responses and true labels. First, we observed cost effect in Llava-1.5-7b, resonating with our Experiment 1 finding about this same VLM (Table 2). Second, we did not observe any competition effects, reproducing prompting-based findings as well as Experiment 1 results. Third, all VLMs showed above-chance match accuracies across conditions and circled objects, surpassing prompting-based results in Figure 3a. Statically, Table 3 confirmed cost effects in Llava-1.5-7b probability-based responses. Overall, Experiment 2 findings were mostly in line with Experiment 1.

5.2.3 Overall match

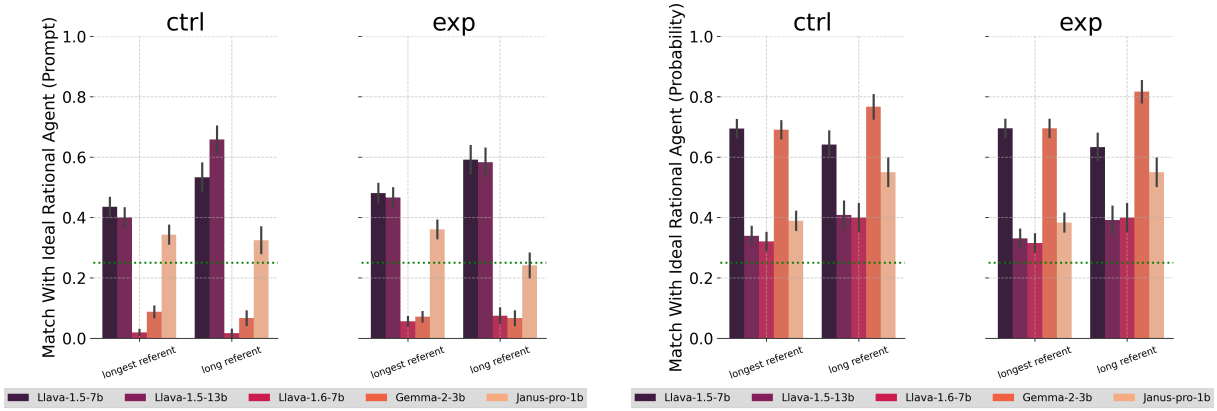
We additionally calculated humans' and VLMs' overall match accuracies and weighted F1 scores aggregating all the trials (Table 4). First, Llava-1.5-7b is the sole VLM that showed good alignment with ideal rational agents in both prompting and probability methods, surpassing human participants' match rates metrics. Second, Gemma is the most sensitive to methods, which showed much higher accuracies and F1 scores in probability method than in prompting method. Third, we found generally higher accuracies and F1 values in probability than in prompting method. Overall, Table 4 echoed the observation in Figures 3a and 3b, suggesting that depending on VLMs, probabilistic measurements can outperform prompting. With explicit instruction, we observed improvements for Llava-1.6-7b prompting and Llava-1.5-7b probability, whereas Gemma (prompting) was biased by the instruction and never chose the ambiguous phrase.

5.3 Experiment 3: Lack of effect

Table 5 demonstrates statistical results of how image features influence VLMs' responses. First, color intensity significantly influenced match odds for Gemma, Janus, Llava-1.5-7b, and Llava-1.6-7b, in prompting or probability methods, suggesting that these models' responses varied depending on whether the image was solid or transparent. Second, we found that for Llava-1.5-13b and Llava-1.6-7b, there were significant variance of VLMs probability responses in the position of objects, suggesting that these VLMs did not remain stable when image features varied. Third, we observed generally stable responses in prompting method than in probability. Fourth, we found color intensity, but not position of objects, more often triggered VLMs' responses variance. With explicit instruction, Gemma and Janus no longer showing significant variance in color, and Llava-1.5-13b and Llava-1.6-7b no longer exhibiting significant variance in position. Meanwhile, adding explicit instruction triggered significant variance in color for Llava-1.6-7b prompting (p=0.015) and Llava-1.5-13b probability (p <0.0001).

5.4 Experiment 4: Model effect

In the first linear mixed effects model (LME), we compared Llava-1.5-13b against Llava-1.5-7b. The results show no difference between the two (Estimate = 0.001, p = 0.958). In contrast, the second LME compared Llava-1.6-7b to Llava-1.5-7b and revealed that Llava-1.6-7b shows lower match scores (Estimate = -0.451, $p < 0.0001$). Together, these results demonstrate that language backbone, rather than parameter size, played a role in matching with ideal rational agent.



(a) Prompting-based metric

(b) Probability-based metric

Figure 3: VLM response match with ideal rational agents. Error bars indicate standard error of the mean. Dotted lines show chance-level match accuracy.

Model	**Color** (p)	**Position** (FDR-p)
Prompting-based response		
Llava-1.5-7b	0.863	0.253
Llava-1.5-13b	0.066	0.453
Llava-1.6-7b	0.246	0.920
Gemma-2-3b	<0.0001	0.306
Janus-1.3b	0.008	0.584
Probability-based response		
Llava-1.5-7b	<0.0001	0.998
Llava-1.5-13b	0.759	<0.0001
Llava-1.6-7b	0.003	<0.0001
Gemma-2-3b	0.015	0.530
Janus-1.3b	0.049	0.770

Table 5: As predicted (sign and magnitude); not as predicted; improved with explicit instruction.

6 Discussion

Some evidence for cost effects Our first two experiments provide evidence that cost effects can emerge in VLMs, although inconsistently across models and measurement methods. In Experiment 1, Janus-pro-1.3b (prompting) and Llava-1.5-7b (probability) showed a higher proportion of short, ambiguous expressions when the costly referents were circled and intended, mirroring the weak directional cost effect observed in humans. Experiment 2 reinforced this pattern: the same models exhibited higher match rates with the ideal rational agent when the intended referent was costly. These results illustrate a first application of ALL as a diagnostic tool for probing cost-related pragmatic behavior in VLMs, suggesting that some models can exhibit strategic cost-sensitive reasoning without relying solely on natural language frequency or lexical priors. One possible explanation for why only Janus and Llava-1.5-7b show this pattern is that both architectures preserve a degree of separation between visual and textual processing, with explicit cross-modal alignment mechanisms (Shu et al., 2025), which may better augment reasoning. Overall, while certain VLMs approximate cost-based pragmatic reasoning in minimal settings, further evidence is needed to establish that this is not only method- or architecture-specific but reflects a robust, model-internal capacity.

Absence of competition effects We found no compelling evidence that VLMs exhibit competition effects driven by ambiguity risk. This contrasts with human sensitivity to ambiguity risk. The absence of competition effects suggests that VLMs do not reliably reason over alternatives in a Gricean way. This finding challenges our hypothesis and aligns with prior work showing that VLMs struggle with pragmatic reasoning despite strong surface-level performance (Verhoef et al., 2024; Cao et al., 2020; Bihani and Rayz, 2025; Misra et al., 2023; Cong and Rayz, 2025; Hua and Artzi, 2024; Vaduguru et al., 2025). The weak pragmatics and lack of robust alternatives-based reasoning also align with broader findings in multimodal evaluations, reporting failures in referential disambiguation, ambiguity handling, among other pragmatic tasks (Qiu et al., 2025; Ma et al., 2025; Testoni et al., 2025; Fu et al., 2025; Shu et al., 2025).

Image features affect some VLMs Contrary to our hypothesis that rational choices should be robust to irrelevant visual cues, color intensity significantly modulated responses in several models, and object position induced variance in larger Llava variants. These effects were more pronounced in probability-based analyses, indicating

that visual salience and low-level perceptual differences likely affect probabilistic pragmatic decision-making. This finding reinforces concerns that VLMs conflate perceptual salience with communicative relevance, a limitation also noted in Liu et al. (2021); Shirnin et al. (2024); Fu et al. (2025).

Model properties As predicted, the sign and magnitude of pragmatic reasoning effects differ across VLMs. Model scale does not influence pragmatic alignment. Increasing parameter size from Llava-1.5-7b to 13b yielded no improvement, whereas replacing the Llama language backbone with Mistral (Llava-1.5 vs. 1.6) substantially reduced alignment with the ideal rational agent. This resonates with previous work advocating for moving beyond sheer scale (Kouwenhoven et al., 2024). Both Llava-1.5 and Llava-1.6 are Vicuna- and CLIP-based, our preliminary findings are also suggestive of limitations in such multimodal architectures. Moreover, consistent improvement in Llava-1.6-7b thanks to explicit instruction in prompting suggests its sensitivity to instructions, but we did not find generalization of such improvement for other VLMs, in fact, explicit instructions triggered VLMs to attend the irrelevant color intensity features, suggesting misalignment with human pragmatic preference (Ma et al., 2025).

The influence of more elaborate prompts For the supplementary prompting experiment, the failure to follow explicit, elaborate instructions regarding feature-invariance suggests that the models conflate perceptual salience with communicative relevance. Rather than ignoring color and position when instructed to do so, the models became fixated on describing these visually salient properties. The results provide another piece of preliminary evidence that the absence of competition effects in VLMs is likely not an artifact of an underspecified prompt, but instead reflects a fundamental reasoning deficit that persists even when the rules of pragmatic behavior are explicitly provided in the input.

Proxy for cost: token length versus phrase length Gemma-based models (e.g., Google's PaliGemma) use a SentencePiece tokenizer that tends to preserve short nonce strings as single tokens (e.g., *zud* and *heth* each map to 1 token, while *zud shegmifankub* maps to 2 tokens). In contrast, Llama- and Mistral-based Llava models apply BPE tokenization, segmenting the same strings into multiple subword units (e.g., *zud* becomes 2 tokens, and *zud shegmifankub* becomes 5 – 6 tokens). Janus, which uses the DeepSeek tokenizer, tokenizes the same phrase *zud shegmifankub* into 3 tokens.

This variation is relevant because *length* as a cost proxy can refer either to human-perceived phrase length or to token count, which more directly reflects a model's representational and processing load. Under this view, a 6-token sequence (e.g., *zud shegmifankub*) is effectively longer for a model than a 2- or 3-token one, even when humans perceive the strings as equally long. Model preferences may therefore track token length rather than phrase length, and this could differ systematically across architectures. A natural follow-up would construct nonce stimuli with matched token lengths across VLMs, enabling cleaner comparison against a common human baseline. Since prior work links greater length to higher cost, aligning token length across models would help clarify whether observed effects are consistent with that literature and whether they hold within VLMs whose behavior more closely mirrors human judgments.

7 Conclusion

VLMs exhibit some cost sensitivity and lack a core component of pragmatic competence: reasoning about ambiguity through competition with alternatives. Our study advocates for theory-driven evaluation frameworks and pragmatically aware VLMs.

8 Limitations

There are several limitations in our work. We operationalized the concept of cost using phrase length, as it has been studied in human behavior. This is an attempt to minimize divergence between human and VLM experiments design, while still allowing valid and feasible comparisons and experiments translation between the two. Future work could explore different approaches for cost manipulations and interactive, listener-based settings to further test whether richer pragmatic behavior can emerge in VLMs.

Tokenizations differ across the VLMs we evaluate. Our analyses rely exclusively on within-model comparisons, with phrase surprisal normalized by token length, and model choices are derived from relative surprisal rankings among competing phrases under a fixed tokenizer. We maintain that absolute differences in token granularity

across models do not necessarily affect the validity of our contrasts or the derived choice-based comparisons reported here. Future studies could systematically examine model-specific tokenization statistics, and show whether the observed cost effects persist when cost is defined by token count rather than phrase length.

Unlike human learners, VLMs in our study do not acquire mappings through iterative correctness feedback. While our structured prompting and one-shot learning design allow us to test VLMs' ability to represent and retain phrase-object associations, we acknowledge that learning mechanisms such as repetition, feedback, and long-term retention are underexplored here. While the primary experiments utilized 12 trials across various renderings, our supplementary qualitative analysis into few-shot interactive prompting was limited to one specific trial configuration. Unlike the main study, which systematically varied color intensity across 768 unique sets, this follow-up focuses on how the models (PaliGemma, Janus, and Llava) respond to a communicative game context within a single trial. Future work could expand our follow-up and systematically vary exposure regimes and incorporate more explicit post-learning comprehension checks within the ALL framework.

The present findings were derived from a highly controlled, narrowly defined task that employed a restricted set of materials. Subsequent research could consider more systematic comparisons between models and human behavior, thereby shedding light on the mechanisms of language learning and use in naturalistic, ecologically valid contexts.

9 Acknowledgments

We thank Brian Buccola for his insights in formal and experimental semantics and pragmatics, Trisha Godara for her help in computational linguistics, and BriGap reviewers for their constructive comments. This project is supported by the College of Liberal Arts, Purdue University.

References

Leon Bergen, Roger Levy, and Noah D. Goodman. 2016. Pragmatic reasoning through semantic inference. *Semantics and Pragmatics*, 9.

Geetanjali Bihani and Julia Rayz. 2025. Learning shortcuts: On the misleading promise of nlu in language models. In *Artificial Intelligence for Design and Process Science*, pages 147–158. Springer Nature Switzerland.

Brian Buccola, Isabelle Dautriche, and Emmanuel Chemla. 2018. Competition and symmetry in an artificial word learning task. *Frontiers in Psychology 9 (2176)*.

Brian Buccola, Manuel Križ, and Emmanuel Chemla. 2021. Conceptual alternatives. *Linguistics and Philosophy*.

Jize Cao, Zhe Gan, Yu Cheng, Licheng Yu, Yen-Chun Chen, and Jingjing Liu. 2020. Behind the scene: Revealing the secrets of pre-trained vision-and-language models. In *Computer Vision – ECCV 2020*, pages 565–580, Cham. Springer International Publishing.

Yan Cong. 2021. *Competition in Natural Language Meaning: The Case of Adjective Constructions in Mandarin Chinese and Beyond*. Michigan State University.

Yan Cong. 2024. Manner implicatures in large language models. *Scientific Reports*, 14(1):29113.

Yan Cong and Julia Rayz. 2025. Language models demonstrate the good-enough processing seen in humans. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 47.

Jennifer Culbertson. 2012. Typological universals as reflections of biased learning: Evidence from artificial language learning. *Language and Linguistics Compass*, 6(5):310–329.

Jennifer Culbertson and Kathryn Schuler. 2019. Artificial language learning in children. *Annual Review of Linguistics*, 5:353–373.

Michael C. Frank and Noah D. Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998–998.

Michael Franke. 2011. Quantity implicatures, exhaustive interpretation, and rational conversation. *Semantics and Pragmatics*, 4(1):1–82.

Michael Franke and Leon Bergen. 2020. Theory-driven statistical modeling for semantics and pragmatics: A case study on grammatically generated implicature readings. *Language*, 96(2).

Michael Franke and Gerhard Jager. 2016. Probabilistic pragmatics, or why bayes' rule is probably important for pragmatics. *Zeitschrift fur Sprachwissenschaft*, 35(1):3–44.

Stephanie Fu, Tyler Bonnen, Devin Guillory, and Trevor Darrell. 2025. Hidden in plain sight: Vlms overlook their visual representations. *arXiv preprint arXiv:2506.08008*.

Bart Geurts. 2010. *Quantity Implicatures*. Cambridge University Press, Cambridge.

H. Paul Grice. 1975. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and Semantics*, volume 3, pages 41–58. Academic Press, New York.

H. Paul Grice. 1989. *Studies in the Way of Words*. Harvard University Press.

Yilun Hua and Yoav Artzi. 2024. Talk less, interact better: Evaluating in-context conversational adaptation in multimodal llms. *arXiv preprint arXiv:2408.01417*.

Roni Katzir. 2007. Structurally-defined alternatives. *Linguistics and Philosophy*, 30(6):669–690.

Tom Kouwenhoven, Max Peeperkorn, and Tessa Verhoef. 2024. Searching for structure: Investigating emergent communication with large language models. *arXiv preprint arXiv:2412.07646*.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. 2018. Emergence of linguistic communication from referential games with symbolic and pixel input. *arXiv preprint arXiv:1804.03984*.

Molly L Lewis and Michael C Frank. 2016. The length of words reflects their conceptual complexity. *Cognition*, 153:182–195.

Fenglin Liu, Xian Wu, Shen Ge, Xuancheng Ren, Wei Fan, Xu Sun, and Yuexian Zou. 2021. Dimbert: Learning vision-language grounded representations with disentangled multimodal-attention. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(1):1–19.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. Llava-next: Improved reasoning, ocr and world knowledge. *arXiv preprint arXiv:2401.17744*.

Haotian Liu, Yuntao Li, Yi Shen, Chunyuan Liu, and et al. 2023. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*.

Ziqiao Ma, Jing Ding, Xuejun Zhang, Dezhi Luo, Jiahe Ding, Sihan Xu, Yuchen Huang, Run Peng, and Joyce Chai. 2025. Vision-language models are not pragmatically competent in referring expression generation. *arXiv preprint arXiv:2504.16060*.

Alexander Martin, Theeraporn Ratitamkul, Klaus Abels, David Adger, and Jennifer Culbertson. 2019. Cross-linguistic evidence for cognitive universals in the noun phrase. *Linguistics Vanguard*, 5(1):20180072.

Kanishka Misra. 2022. minicons: Enabling flexible behavioral and representational analyses of transformer language models. *arXiv preprint arXiv:2203.13112*.

Kanishka Misra, Allyson Ettinger, and Julia Rayz. 2020. Exploring bert's sensitivity to lexical cues using tests from semantic priming. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4625–4635.

Kanishka Misra, Allyson Ettinger, and Julia Taylor Rayz. 2021. Do language models learn typicality judgments from text? *arXiv preprint arXiv:2105.02987*.

Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. Comps: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2928–2949.

Yasamin Motamedi, Marieke Schouwstra, Kenny Smith, Jennifer Culbertson, and Simon Kirby. 2019. Evolving artificial sign languages in the lab: From improvised gesture to systematic sign. *Cognition*, 192:103964.

Dojun Park, Jiwoo Lee, Hyeyun Jeong, Seohyun Park, and Sungeun Lee. 2024. Pragmatic competence evaluation of large language models for the Korean language. In *Proceedings of the 38th Pacific Asia Conference on Language, Information and Computation*, pages 256–266, Tokyo, Japan. Tokyo University of Foreign Studies.

Tiago Pimentel, Clara Meister, Ethan Wilcox, Kyle Mahowald, and Ryan Cotterell. 2023. Revisiting the optimality of word lengths. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 2240–2255.

Elena Pitta, Tom Kouwenhoven, and Tessa Verhoef. 2025. Probing vision-language understanding through the visual entailment task: promises and pitfalls. In *Proceedings of the 2nd LUHME Workshop*, pages 74–83.

Ciyang Qing and Michael C. Frank. 2014. Gradable adjectives, vagueness, and optimal language use: A speaker-oriented model. In *Proceedings of Semantics and Linguistic Theory (SALT) 24*, volume 24, pages 23–41.

Linlu Qiu, Cedegao E Zhang, Joshua B Tenenbaum, Yoon Kim, and Roger Levy. 2025. On the same wavelength? evaluating pragmatic reasoning in language models across broad concepts. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 19924–19946.

Jessica Rett. 2015. *The Semantics of Evaluativity*. Oxford University Press.

Jessica Rett. 2019. Manner implicatures and how to spot them. *International Review of Pragmatics*, in press.

Emmanuelle Salin, Badreddine Farah, Stéphane Ayache, and Benoit Favre. 2022. Are vision-language transformers learning multimodal representations? a probing perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11248–11257.

Alexander Shirnin, Nikita Andreev, Sofia Potapova, and Ekaterina Artemova. 2024. Analyzing the robustness of vision & language models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2751–2763.

Dong Shu, Haiyan Zhao, Jingyu Hu, Weiru Liu, Ali Payani, Lu Cheng, and Mengnan Du. 2025. Large vision-language model alignment and misalignment: A survey through the lens of explainability. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1713–1735, Suzhou, China. Association for Computational Linguistics.

Andreas Steiner, André Susano Pinto, Michael Tschannen, Daniel Keysers, Xiao Wang, Yonatan Bitton, Alexey Gritsenko, Matthias Minderer, Anthony Sherbondy, Shangbang Long, and 1 others. 2024. Paligemma 2: A family of versatile vlms for transfer. *arXiv preprint arXiv:2412.03555*.

Alberto Testoni, Barbara Plank, and Raquel Fernández. 2025. Racquet: Unveiling the dangers of overlooked referential ambiguity in visual llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 23638–23658.

Saujas Vaduguru, Yilun Hua, Yoav Artzi, and Daniel Fried. 2025. Success and cost elicit convention formation for efficient communication. *arXiv preprint arXiv:2510.24023*.

Tessa Verhoef, Kiana Shahrasbi, and Tom Kouwenhoven. 2024. What does kiki look like? cross-modal associations between speech sounds and visual shapes in vision-and-language models. *arXiv preprint arXiv:2407.17974*.

Elspeth Amabel Wilson. 2017. *Children's development of Quantity, Relevance and Manner implicature understanding and the role of the speaker's epistemic state*. Ph.D. thesis, University of Cambridge.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, and 1 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. 2024. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *Preprint*, arXiv:2410.13848.

A Prompt excerpts

Learning in one-shot format:
The label for this object is <label>.

Learning in few-shot format:

Note: the following prompt was presented three times, and each paired with an image where the object is in a different color and in a different position.

The label for the circled object is <label>. The color, size, and position of the object do not determine its name; only its shape matters.

Gaming without explicit instruction (default):
Identify the label for the circled object in the image from the following options: <label>zud</label>, <label>zud vun</label>, <label>zud shegmifankub</label>, <label>heth</label>.

Gaming with explicit instruction:
Identify the label for the circled object in the image from the following options: <label>zud</label>, <label>zud vun</label>, <label>zud shegmifankub</label>, <label>heth</label>. Respond with only the label without explaining. Choose the option that concisely and uniquely identifies the circled object.

Gaming with explicit instruction and interactive context:
You are playing a game with a Martian. The Martian sees the same three objects you see. You must send them a label so they can correctly pick the object you have circled. Identify the label for the circled object. Note that the size and position of the object do not determine its name; only its shape matters. In this Martian language, short phrases are preferred over long phrases if both are true. Your goal is to use the shortest possible label that uniquely identifies the target. Choose from the following options: <label>zud</label>, <label>zud vun</label>, <label>zud shegmifankub</label>, <label>heth</label>. Respond with only the label without explaining. Which label do you send?

B Supplementary figures

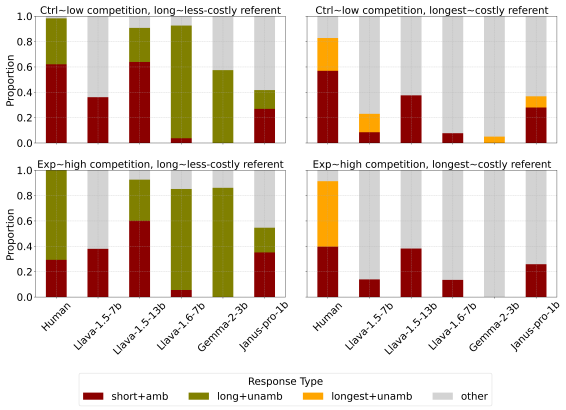


Figure 4: VLMs-prompt responses in the testing phase with explicit instruction prompt, grouped by condition and circled object.

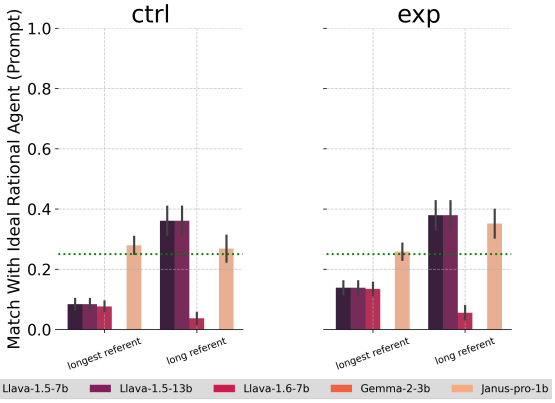


Figure 6: VLMs prompting responses match with ideal rational agents, with explicit instruction prompt. Error bars: standard error of the mean. Dotted line: chance-level match accuracy.

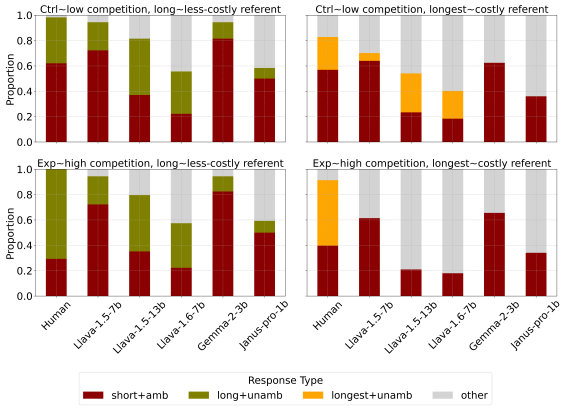


Figure 5: VLMs probability responses in the testing phase with explicit instruction prompt, grouped by condition and circled object.

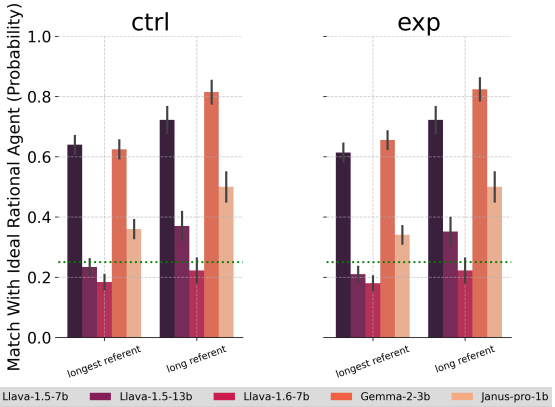


Figure 7: VLMs probability responses match with ideal rational agents, with explicit instruction prompt. Error bars: standard error of the mean. Dotted line: chance-level match accuracy.

Using the Mimi codec for metalinguistic representations

Artem Saloev[1], Erin Pacquetet[2], Nicolas Ballier[1]
[1]ALTAE, Université Paris Cité, F-75013 Paris, France
[2]SCIAM, 10 rue de Penthièvre, F-75008 Paris, France
artem.saloev@gmail.com, erin.pacquetet@sciam.fr,
nicolas.ballier@u-pariscite.fr

Abstract

Codec-based audio language models are developing, but little explainability research has been dedicated to the representation of this type of speech tokenisation. In this paper, we focus on the dictionary of 2048 tokens used in Mimi's semantic token codebook, the neural codec of the Moshi language model (Défossez et al., 2024). We show that the ABX experiment carried out with Mimi fails to capture the mapping of the semantic tokens to phone realisations. By realigning Mimi's representations to the TIMIT corpus transcriptions (Garofolo et al., 1993), we show that the 2048 tokens IDs of the semantic codebook map to quadphone, triphone, biphone, phone and subphone realisations. We used the TIMIT transcriptions as evidence of the validity of the allophone-based representations of these 80ms semantic token representation and examine some of the theoretical consequences for the tokenisation of speech at allophone and subphonemic level.

1 Introduction

Codec-based audio language models have recently emerged as a way to apply Transformer-style sequence modelling to speech by first turning the waveform into a stream of discrete codec tokens. A neural audio codec compresses the signal using multiple codebooks, or layers (typically via residual vector quantisation, RVQ), yielding a small number of tokens per frame; a Transformer is then trained to predict token sequences, and a decoder reconstructs the waveform from those predictions. Working with discrete audio tokens brings benefits: stable long-context modelling, controllable bit-rate and latency (via tokens per second and the number of codebooks used), and, in some designs, faster generation by predicting coarse levels first and refining finer details in parallel. This type of technology is increasingly used for conversational agents or speech-to-speech translations (Labiausse et al., 2025) but little is known about how voices and speech features are encoded in these neural audio codecs. As conversational agents aim at real-time interactions and reducing latency, this has led to the design of codecs with limited frame rates (in our case study, 80 ms frames). While this design is undoubtedly motivated by engineering considerations, the decision to discretise a naturally continuous acoustic signal raises a fundamental question : do the learned audio tokens partition the signal in ways that align with the categories posited by phonology (and, more broadly, phonetics)? Answering this question is central both to analysing these models' internal representations and to assessing whether there is an alternative way to model the phon* level.

Our goal in this paper is to test how far the two representations coincide. We focus on Mimi, a recent streaming neural codec that produces multi-level token sequences suitable for spoken-dialogue modelling. Using TIMIT (the standard testbed for the phonetics of American English and its computational modeling), with its time-aligned phonetic transcriptions, we ask whether Mimi's codebook tokens can be mapped onto the manually identified phonemes. This seemingly simple mapping raises immediate practical difficulties: token frames and phoneme boundaries are not naturally aligned; tokens have a fixed frame rate, whereas phoneme durations vary with context; and some information relevant to segment identity may be distributed across levels.

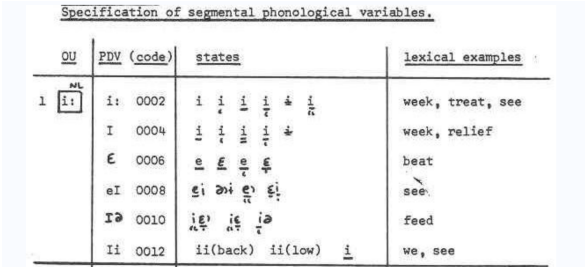
The audio language model Moshi generates and encodes speech as discrete audio tokens using a neural audio codec called "Mimi"[1]. The Mimi codec quantizes audio waveforms into these tokens. It operates at a frame rate of 12.5 Hz (frames per second). This means that each "temporal frame" of audio processed by Moshi, and thus each time

[1]https://huggingface.co/kyutai/mimi

Third Workshop on the Bridges and Gaps between Formal and Computational Linguistics, pages 63–73

step for the Temporal Transformer, corresponds to an 80-millisecond (ms) slice of audio (1 second / 12.5 Hz = 0.08 seconds or 80 ms). Moshi's design integrates the generation of both acoustic and semantic tokens within Mimi. Rather than having a completely separate semantic token codebook, Mimi's architecture is designed to distill semantic information into a specific part of its quantization process, effectively creating semantic-rich tokens. Mimi discretizes waveforms into audio tokens from a self-supervised speech model, WavLM (Chen et al., 2022), into tokens. This allows for streaming encoding and decoding of semantic-acoustic tokens. WavLM (Chen et al., 2022) was trained on English with Gigaspeech (Chen et al., 2021) and on the 23 languages of the European Parliament with the VoxPopuli Dataset (Wang et al., 2021b).

Our approach is bottom-up in spirit. We assume each 80 ms interval of audio is encoded by a single token in the Mimi codec encoder. We map codebook_0 tokens (the semantic token codebook) to each 80 ms interval and re-align these intervals with the phone-aligned data of the TIMIT corpus. Because an 80 ms token may not capture the phone level, we cannot rely on the Phone Error Rate (PER) metric (an alternative reason being we have a dictionary of 2048 tokens for a 72 phone set). Based on standard studies of phone durations in English speech (Keating et al., 1992; Byrd, 1992; Keating et al., 1994), a plausible hypothesis is that an 80ms time frame is likely to capture triphones, biphones, phones and "subphones" (in particular, fragments of vowels). We partially replicate previous analyses of codecs that have been used for example with SpeechTokenizer (Zhang et al., 2024): we have used the TIMIT training data to match the Mimi semantic codebook token representations to TIMIT phonetic data transcriptions at phone level, word level and utterance level. We found the word-aligned transcriptions to be the most relevant to capture the gist of the semantic token representations, providing cogent representations of word allophone networks proposed in the early days of the TIMIT analyses. We have also used this available resource on English with the belief that the semantic codebook representations could be used to partially replicate previous phonetic analyses on the TIMIT data (Keating et al., 1992; Byrd, 1992; Keating et al., 1994; Byrd, 1994) in the future, leading to extracting features for phone recognition tasks (Young and Woodland, 1994), dialectology (Clopper and Pisoni, 2004b,a) and computational


Specification of segmental phonological variables.

OU	PDV (code)	states	lexical examples
1 NL i:	i: 0002	i i i i i i	week, treat, see
	I 0004	i i i i i	week, relief
	ɛ 0006	e e e e	beat
	eI 0008	ei əi ei ei	see
	Iə 0010	iɛ iɛ iə	feed
	Ii 0012	ii(back) ii(low) i	we, see

Figure 1: Levels of granularity in the Tyneside Linguistic Survey, from (Jones-Sargent, 1985)

dialectology (Miller and Trischitta, 1996). For example, when investigating the phonetic realizations of Geordie (Newcastle English) in the 1960s for the Tyneside Linguistic Survey, linguists tried to annotate different levels of analysis. They considered as the major arch category the OU, the overall unit, and this more or less corresponds to the phonological level with the assumption that this abstraction corresponds to the class of realizations. In that respect, the mapping of the phonological category was not single-handedly over to the notion of phonetic realizations as diacritics, but was conceived as a form of several types of realizations. The PDV, the Putative Disystemic Variable, which assumes that a form of sub-categorization can be proposed to account for the different states, would correspond to individual realization, for example, as encountered in some specific lexical examples. So it might be the case that codec tokens could be very similar to the code that was associated in the transcription of the corpus that was coded on the basis of the four-digit code (see Figure 1). The (MIMI) codec tokens, if consistently encoding a portion of the acoustic signal bijectively could be used to encode the extreme level of granularity of the phonetic representation. Another potential application of this property would be the automatic annotation of code-switching.

There are two main contributions in our paper. First, we establish a preliminary inventory of Mimi's "semantic" tokens observable in the TIMIT data with a method and code that can be replicated with the other 23 languages used to train the Mimi NAC. Second, we discuss the theoretical consequences of this type of speech tokenisation and evidence some of the potential benefits for the phonetic community of this type of "semantic" (indeed, allophonic) tokens.

The rest of the paper is organised as follows: Section 2 presents the main NACs that have been

previously been investigated. Section 3 explains how we have analysed the Mimi codebook and how we organised the mapping of the semantic tokens with the TIMIT training data. Section 4 presents our results, Section 5 discusses these results and Section 6 suggests future research. Section 7 concludes.

2 Previous Research

2.1 Neural Audio Codec Investigation

Previous research comparing NACs has tested neural codecs on text-to-sound task (Ye et al., 2025), such as their ability to generate speech from texts using text-to-speech applying UTMOS as a metrics, a speech MOS (Mean Opinion Score) predictor (Saeki et al., 2022). They mostly investigate the interpretability of NAC at the decoding end, comparing the NAC outputs with human judgment. Other research focused on understanding how speech attributes like content, identity, and pitch are encoded within these codecs by proposing a two-step analysis and synthesis approach (Sadok et al., 2025). Though focusing on SpeechTokenizer and BigCodec, (Sadok et al., 2025) partially discusses Mimi as one of four contemporary NACs studied for their ability to compress and decompress audio while maintaining a high level of quality.

(Halimeh et al., 2025) investigates how the latent representations learned by neural speech codecs relate to quality but also demonstrates that there is not a direct, interpretable relationship between quantized latents and the qualities of the raw speech signal, further supporting the case that neural speech codec representations are less interpretable than raw signals. Lastly, (Mousavi et al., 2025) proposed a typology of codecs and an across-the-board benchmark of codecs, extending the analysis of speech generation to audio and music generation.

2.2 Mimi ABX experiment (Défossez et al., 2024)

When releasing the Mimi codec, (Défossez et al., 2024) tested the potential phonetic representation of its semantic tokens using ABX discrimination tasks. The Phonetic Discriminability Evaluation, measured by the ABX error rate (Schatz et al., 2013), was used to characterize the phonetic discriminability of the semantic token's representation space. This metric compares distances between embeddings of different instances of the same triphone (e.g., "big") versus a minimally different contrastive triphone (e.g., "beg") from the same speaker. A lower ABX error rate indicates better phonetic discriminability. We argue this procedure fails to capture what 80 ms of speech may represent phonetically. The ABX assumes phonemic representation but with an 80 ms slice of speech, we could be below the phoneme level in an elongated vowel, and with many consonants, we are at the cluster level. We believe the time frame associated to the Mimi tokens cannot be consistent with a phonemic representation, but is very likely to capture complex co-articulation phenomena: mutual influences of neighbouring sounds that are well-established in the phonetics/phonology literature. [2] The token interval may not correspond to the phone or word interval but still provide consistent representation (encoding) of acoustic and phonetic properties of the aligned material (phone, diphone, triphone at the phonetic level; allophone, cluster or syllables at the phonological level).

3 Material and Methods

3.1 Models

We used part of the inference code made public by Kyutai and extracted the 8 codebooks with the Mimi model available from hugging face.[3] For this paper, we only investigated codebook_0, the codebook supposedly corresponding to the "semantic" tokens, specifically trained for this purpose (Défossez et al., 2024). As explained in (Défossez et al., 2024), during training, embeddings from WavLM (Chen et al., 2022) are distilled into a single vector quantizer which produces semantic tokens and that is combined with separate acoustic tokens for reconstruction. In other words, the model builders claim that semantic information is more likely to be captured for this first codebook (codebook_0): the tokens of this codebook are learnt by distillation from WavLM, a large-scale multi-lingual unlabelled audio dataset consisting of over 400k hours of audio in 23 languages collected from the European Parliament sessions, committee meetings, and other events. The audio tokens are running at 12.5Hz and with a bitrate of 1.1kbps. The semantic tokens are combined with separate acoustic tokens for reconstruction in the decoding phase. (Bal-

[2] Last, this moving windowed lens on the speech signal is very sensitive to phone alignment and the Phonetic Discriminability Evaluation dataset used for the evaluation does not seem to have been manually checked.

[3] https://huggingface.co/kyutai/mimi.

lier et al., 2026) confirmed that these codebooks, (codebook_1) to (codebook_7) partially captured prosodic features such as duration, pitch and glissando.

3.2 Data

We used the official training and test partitions of the TIMIT Acoustic-Phonetic Continuous Speech Corpus (Garofolo et al., 1993), a standard benchmark for phonetic and speech recognition research. The TIMIT dataset contains of speech recordings from 630 speakers covering eight major American English dialect regions, with each speaker reading 10 phonetically-balanced sentences selected to maximize coverage of phonemes and phonetic contexts.

Data splits. The corpus is divided into predefined training and test sets. The training set consists of 4,620 utterances from 462 speakers, while the test set includes 1,344 utterances from 168 speakers, ensuring balance across dialect regions and speaker demographics.

Data structure. Each utterance is provided as a 16kHz, 16-bit linear PCM waveform file, accompanied by time-aligned transcriptions at the sentence, word, and phoneme levels.

3.3 Method

First, we encoded the audio files from the TIMIT training and testing data into Mimi codebooks[4]. We then matched sequences of words and phones to their corresponding token encoding, focusing on the semantic tokens represented in codebook_0. Because each 80 ms interval of audio is encoded with a single token, the alignment between phones, words and token rarely coincides. Some words are spoken throughout multiple 80 ms intervals, while some tokens are encoding for multiple phones at a time. Table 1 shows an example of the time alignment mismatch between phone, word and token (codebook) level.

Because the tokens produced are time-dependent, token time stamps do not align with phonetic transciptions and the pronunciation of words. We consequently decided to report three levels of analysis for the Mimi semantic tokens: at phone level (interpreting the partial matching of token IDs with TIMIT phone set labels) at word level (interpreting the transcriptions of words in tokens) and at utterance level (trying to predict the different regions of the TIMIT data based on the transcriptions of utterances into tokens).

[4]We used the HuggingFace implementation and the Kyutai github. These supplementary resources, as well as code for reproducibility (including realigned TextGrids and our codebooks), will be publicly released via Github upon publication.

token	start (in ms)	end (in ms)	phone	word
888	640	720	epilnlaw	now
495	720	800	aw	now
1586	800	880	awlix	now/ adjourned

Table 1: Example of token to phone to word alignment

3.4 Metrics

We analysed the relevance of the token sequences for transcriptions at phone level, word-level and utterance level. At the phone level, we could not use the more standard Phone Error Rate (PER) metric, which would compare a predicted phone sequence (from Mimi) to the gold standard TIMIT phonetic alignments. Instead, we replicated the method followed for the analysis of the SpeechTokenizer codec (Zhang et al., 2024) and computed the equivalent of the Phone-Normalized Mutual Information (PNMI) (Hsu et al., 2021). In other words, we investigate how pure the correspondence between TIMIT phone sequences and the time frames of the tokens. At the word-level, we used the training set of the TIMIT data and computed the percentage of words that were transcribed in the testing set with the same sequences of tokens as in the training data. At the utterance level, we used accuracy for the dialect classification task.

4 Results

4.1 Phone level

We first report the correspondences we observed between the assumed time frames of the semantic tokens and the most frequent co-occurring TIMIT phone or sequences of phones observed. In the training set of the TIMIT Corpus, 1793 tokens IDs were identified out of the 2048 dictionary IDs of the codebook. The utilization rate of the Mimi semantic codebook (87.54%) is much more optimal than the one reported (Zhang et al., 2024) for the EnCodec NAC (Défossez et al., 2023). Table 2 summarises the main correspondences between the semantic token time frames and the TIMIT phones or phone sequences reported in the TIMIT corpus

(majority) TIMIT correspondence	n
allophones/subphones	570
diphones	980
triphones	232
quadphones	11

Table 2: Distribution of TIMIT phone (sequences) matching Mimi semantic tokens

alignments.

Assuming the correspondence is between semantic token IDs and possible sequences of TIMIT phones, we computed the equivalent of the Phone-Normalized Mutual Information (PNMI) (Hsu et al., 2021), which computes the purity of the matching between the majority vote and the other candidates observed in the "receptive field" of the tokens. We obtained a value of 0.66, which ranks much above HuBERT (0.43), EnCodec (0.28) and slightly below SpeechTokenizer (0.71) but still similarly suggests that the semantic distillation process is effective (Zhang et al., 2024).

4.2 Word-to-Token Re-transcription Predictability

At word-level, using the word timestamps of the TIMIT corpus, we examined all the token retranscriptions of the words that were both present in the training set and in the testing set. For the 2,378 words present in the two sets, only 85 were transcribed differently in the testing set. Only 3.57 % of the token sequences corresponding to the transcriptions of words were not predicted. It is likely that dialectal features could explain the observed variation in token sequences. Many frequent words having allophonic variations, such as words having weak forms (eg *a, her, would*), and the very linguistic token chosen to design the comparable subcorpus SA, "*greasy*" were found to have different token transcriptions in the training and testing sets.

4.3 Utterance-Level Analysis

Assuming that the sequences of semantic tokens can be used as allophonic transcriptions of TIMIT data, we used the utterances read by all the speakers and tried to predict the dialect region of the speaker on the basis of the clusters of token IDs found in the transcription of each sound file. Assuming the eighth region is more difficult to predict as it corresponds to mobile speakers, we carried out a seven-class classification task. We created a sparse matrix for the feature matrix that does not assume that all codebook_0 vectors have the same length (Table 3). Each row in the dataset may have a codebook_0 vector (token list) of a different length. The feature matrix is built by creating a sparse binary matrix where each unique token across the entire dataset becomes a column, and a value of "1" indicates that the token is present for a given sample. Sentences may have differing numbers of tokens, but the resulting matrix will always have the same number of columns (all unique tokens), with each row containing as many "1"s as the number of tokens present for that sample. The method is more computer-intensive but robust to variable-length codebook_0. We used a 80-20 split for a random forest model. The random forest model significantly outperformed the no-information rate, $p < .001$, with an accuracy of 32.63% (95% CI [26.69%, 39.01%]) and a Cohen's $\varkappa = 0.20$.

	DR1	DR2	DR3	DR4	DR5	DR6	DR7
DR1	2	0	0	0	0	0	0
DR2	8	17	12	4	3	6	10
DR3	1	7	12	5	4	4	11
DR4	0	4	6	13	8	0	4
DR5	4	0	1	16	16	0	2
DR6	2	1	1	0	3	5	1
DR7	2	11	8	2	5	3	12

Table 3: Confusion matrix of the Random Forest prediction of the TIMIT dialect region (Reference as columns, predictions as rows

4.4 Entropy-Based Token–Phoneme Analysis

To refine the interpretation of these observations, we incorporated an entropy-based analysis of token–phoneme relationships using the TIMIT corpus. Importantly, TIMIT contains phonetically balanced sentences that are read by all speakers, including two shared utterances (SA1 and SA2). Although these sentences are not identical, they provide a controlled setting with consistent speaker coverage and comparable phonetic content.

The results reveal a strong asymmetry between the conditional distributions $P(\text{phoneme} \mid \text{token})$ and $P(\text{token} \mid \text{phoneme})$. Specifically, tokens exhibit low entropy and are strongly associated with a dominant phoneme, whereas each phoneme corresponds to a broad distribution of competing tokens.

To further investigate token stability, we compare token distributions for phonemes that occur in both SA1 and SA2. Despite being produced by the same set of speakers, we observe high Jensen–

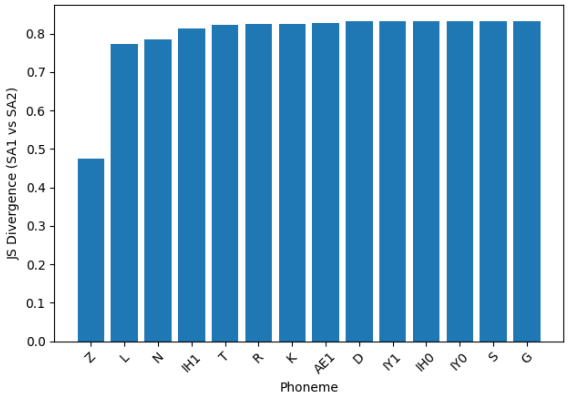


Figure 2: Jensen–Shannon divergence between token distributions for shared sentences (SA1 vs SA2) across phonemes. High divergence values indicate substantial variation in token assignments across different phonetic contexts.

Shannon divergence between token distributions for many phonemes, often exceeding 0.8 (see figure 2). This indicates substantial variation in token assignments across different phonetic contexts.

This many-to-one mapping provides quantitative support for the hypothesis that Mimi tokens encode finer-grained distinctions than phonemic categories. Rather than forming a discrete phoneme inventory, the first codebook partitions the acoustic space into clusters reflecting context-dependent realizations.

This variability cannot be fully explained by phonemic distinctions alone. Even when considering the same phoneme across comparable utterances, token distributions vary significantly, indicating sensitivity to contextual and coarticulatory factors. This result supports an allophonic interpretation of the tokens, where each phoneme is realized through multiple context-dependent variants encoded as distinct tokens.

5 Discussion

5.1 Towards Subphonetic and Allophonic Representations

The results suggest that Mimi's semantic codebook does not encode phonemes as discrete symbolic units, but rather captures finer-grained acoustic-phonetic patterns. This observation is consistent with the idea of "bypassing transcription" (Bird, 2020), where linguistic structure is inferred directly from the speech signal without relying on predefined phonemic categories.

A useful parallel can be drawn with subphonetic modeling in traditional speech recognition. In Hidden Markov Model (HMM)-based systems (Rabiner, 2002), context-dependent states (often referred to as senones) are used to capture variation in phoneme realizations across different contexts. These units do not correspond directly to phonemes, but rather to clustered acoustic realizations conditioned on phonetic environment (Hinton et al., 2012), (Wang et al., 2023).

Similarly, the Mimi tokens appear to function as context-dependent acoustic units. The entropy asymmetry and the high divergence observed across contexts indicate that a single phoneme corresponds to multiple token realizations, depending on coarticulation, prosody, and speaker-specific factors. It may suggest that the semantic codebook implicitly encodes a structured space of allophonic variation, and Mimi tokens can be interpreted as forming an emergent inventory of subphonetic units, situated between continuous acoustic representations and discrete phonological categories.

6 Potential Applications for Phonetic Modeling

In this section, we compare the speech tokenization operated by Mimi to previous subphonetic modeling representations, especially for context-dependent subphonetic representations (Hwang and Huang, 1992).

6.1 Revisiting Allophone Networks

The "allophone network"(Cohen et al., 1987) represented in Figure 3 for the word "*water*" shows the visualization of the main allophones expected for this word. A single phone [w] represents the initial consonant. The vowel opposes two possible realizations or trajectories in the network, the "*bob*" vowel or the "*bought*" vowel. The possibility of flapping is indicated for the medial consonant by DX and two possible allophones are noted for the final vowel, the central rhotic vowel as in *bird* or the reduced vowel with no rhoticity found in northern varieties (schwa). This networking of the allophone variants can be replicated with the word *greasy* using the TIMIT phoneset representations or with the neural audio encoder tokens with many more details as evidenced in Figure 4. Six positions can be observed (compared to 9 in the token/subphonemic representation) for the TIMIT phone set representation of the pronunciations of "*greasy*" in the training data.

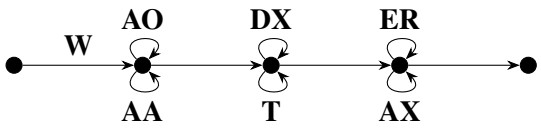


Figure 3: Allophone network for the word *water* (Cohen et al., 1987)

6.2 Speech Tokenizers as Metalinguistic Representations

The behaviour of Mimi's semantic codebook suggests parallels with earlier approaches to subphonetic modeling in speech recognition. Rather than encoding discrete phoneme-like symbols, the tokens appear to capture context-dependent acoustic-phonetic realizations shaped by coarticulation and phonetic environment. This interpretation is consistent with the entropy asymmetries and token-to-phone correspondences observed in our analyses.

A useful comparison can be drawn with senones in Hidden Markov Model (HMM)-based ASR systems (Hwang and Huang, 1992). Senones were introduced as context-dependent subphonetic units designed to model variation in phoneme realizations across different acoustic environments. Similarly, Mimi tokens appear to function as distributed acoustic units that encode finer-grained variation than phoneme-level categories alone.

6.3 Cross-Linguistic Evidence from VoxPopuli

To extend the analysis beyond English, we examined token distributions across language families using the VoxPopuli corpus (Wang et al., 2021a). Preliminary observations on Mimi representations across the 27 European Union languages suggest that token distributions tend to cluster according to language families, as illustrated in Table 4. This grouping indicates that the first codebook captures structured phonetic variation that is shared across related languages.

Language Family	Number
Germanic	85
Romance	223
Slavic	187
Finno-Ugric	313
Baltic	368
Hellenic	520
Semitic	378

Table 4: Distribution of VoxPopuli languages by family.

These findings provide additional support for the allophonic hypothesis. If tokens were purely phonemic, one would expect more language-specific and discrete mappings. Instead, the observed clustering across language families suggests that tokens encode phonetic properties that generalize across languages, such as articulatory or acoustic similarities shaped by shared phonological systems.

However, these results also highlight an important limitation: the relationship between phonetic units and Mimi tokens is not strictly one-to-one. Rather than forming a stable inventory of phoneme-like units, the codebook appears to implement a distributed representation in which phonetic entities are mapped onto multiple context-dependent tokens. This observation is consistent with the entropy-based analysis, which shows that each phoneme corresponds to a broad set of competing tokens.

6.4 Typologically Related Results

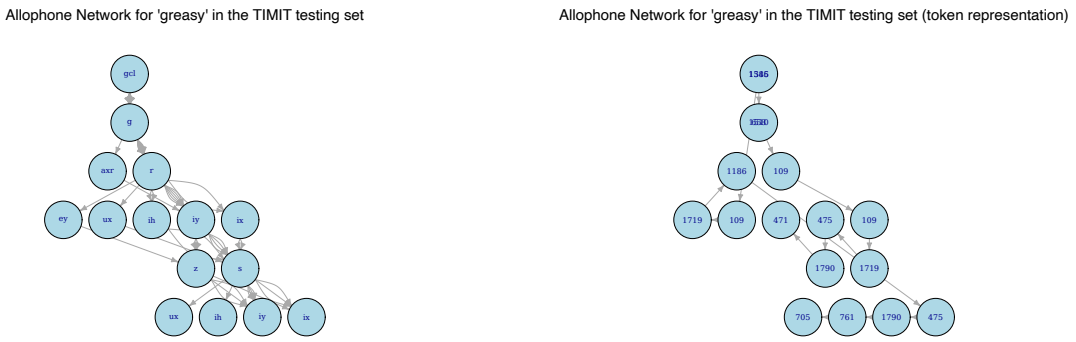
Figure 5 summarizes three complementary metrics: Shannon entropy, token coverage, and maximum token probability, computed for each codebook across typologically related language groups.

First, the entropy patterns are highly consistent across language families. All groups exhibit increasing entropy from codebook_0 to higher codebooks, indicating progressively more uniform token usage. Crucially, this trend is not language-specific but shared across different families, suggesting that the structure of the representation is largely language-independent.

Second, the number of missing tokens decreases sharply across codebooks, reflecting improved coverage of the acoustic space. While codebook_0 shows substantial sparsity, higher codebooks achieve near-complete coverage across all language families. This indicates that the representational capacity of the codec is distributed across multiple codebooks rather than localized in a single layer.

Third, the maximum token probability decreases across codebooks, showing that token distributions become less dominated by a small number of units. This again supports the idea of a more distributed and fine-grained encoding at deeper levels of the representation.

Moreover, the fact that token distributions evolve similarly across codebooks for all language families indicates that phonetic structure is not encoded in isolation within codebook_0, but emerges



(a) TIMIT phone set representation

(b) Mimi token representation

Figure 4: Comparison of the TIMIT phone transcriptions and Mimi semantic token representations of the allophone network for the pronunciations of the word *greasy* in the TIMIT training data

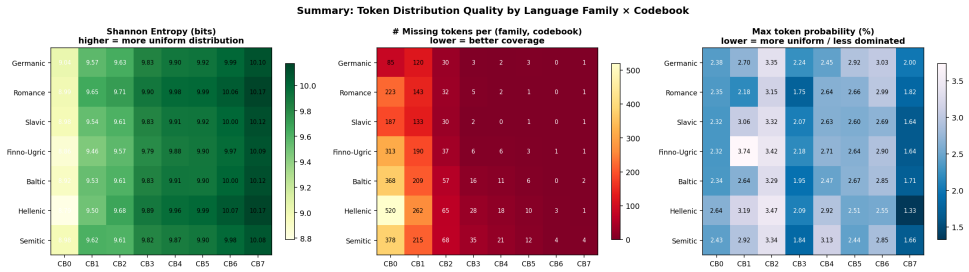


Figure 5: Token distribution quality across language families and codebooks. Left: Shannon entropy (higher indicates more uniform distributions). Middle: number of missing tokens (lower indicates better coverage). Right: maximum token probability (lower indicates less dominance).

from the interaction of multiple representational layers. This reinforces the view that Mimi implements a distributed and context-sensitive encoding of speech, in which phonetic units are not mapped one-to-one onto discrete tokens but are instead represented through combinations of tokens across codebooks.

6.5 Token Distribution Across Codebooks and Language Families

To further investigate the structure of the Mimi representations, we analyze the distribution of token probabilities across codebooks and language families. Figure 6 presents the probability density of tokens for each codebook (CB0–CB7) and language family.

A clear structural difference emerges across codebooks. The earliest codebooks (CB0–CB2) exhibit highly peaked distributions, where a small number of tokens dominate the representation. This indicates that these codebooks capture more selective and possibly lower-level acoustic or phonetic features. In contrast, later codebooks (CB4–CB7) display flatter and more uniform distributions, suggesting a broader and less specialized use of tokens.

6.6 Observations on Greek Phonetic Specificity

Although phoneme-level alignment is not yet incorporated, preliminary observations can be made regarding Greek, the sole representative of the Hellenic language family in the dataset. Greek is known for its relatively stable vowel system and distinct consonantal contrasts, including fricatives such as /x/ and /ɣ/, which are less common in many other European languages (Mennen and Okalidou, 2006).

In the token distributions, the Hellenic group exhibits patterns broadly consistent with other language families, but with slight variations in peak structure and dispersion. These differences may reflect the influence of language-specific phonetic features, particularly in the realization of fricatives and vowel quality.

Figure 6: Token probability density across codebooks (CB0–CB7) and language families. Each subplot shows the distribution over token indices, highlighting differences in sparsity and dominance across codebooks.

7 Conclusion

In this paper, we compared the TIMIT phone alignment with Mimi's first codebook representing semantic tokens and measured the consistency of the mapping we produced. It remains to be seen whether the acoustic tokens (the 7 other parallel codebooks used to regenerate speech with the semantic codebook) could similarly be reconnected to properties of the speech signal such as suprasegmental features. The growing number of downstream tasks relying on this NAC makes it an object of research worthy of interest: the Mimi codec has been used with the Moshi conversational agent and Hibiki (Labiausse et al., 2025) for speech translation. It has more recently been adopted for Sesame's Maya conversational speech model[5].

An important limitation of our work is we did not fully investigate gender effect and potential duplicates of tokens, which could be attributed to different formant values for male and female voices. Traditional signal processing methods could be applied to the latent space to extract meaningful features, such as energy, entropy, or frequency content, which might reveal important characteristics of the learned representations. We have not used some of the acoustic correlates of the codes. For instance, we could exploit mid-temporal value formant extractions to assign the prototypical value of vowels to their corresponding token IDs.

Finally, these results open a perspective that could be described as computational creolistics: the investigation of how speech representations emerge through the sharing and recombination of tokens across speakers and languages. Rather than forming fixed phonological categories, Mimi tokens appear to function as indexical units, whose distribution reflects both shared acoustic structure and context-dependent variation, paving the way for a computational metalinguistic representation of allophonic classes.

Acknowledgement

We thank the three reviewers for their constructive feedback. This publication has emanated from research supported in part by a 2021 research equipment grant from the Scientific Platforms and Equipment Committee (PAPTAN project), under ANR Grant Number ANR-18-IDEX-0001 (Financement IdEx Université de Paris).

[5] https://huggingface.co/sesame/csm-1b

References

Nicolas Ballier, Artem Saloev, Erin Pacquetet, and Taylor Arnold. 2026. Is Prosody Encoded in the Neural Audio Codec Mimi? In *Speech Prosody 2026*, pages 569–573.

Steven Bird. 2020. Decolonising speech and language technology. In *Proceedings of the 28th international conference on computational linguistics*, pages 3504–3519.

Dani Byrd. 1992. Preliminary results on speaker-dependent variation in the timit database. *The Journal of the Acoustical Society of America*, 92(1):593–596.

Dani Byrd. 1994. Relations of sex and dialect to reduction. *Speech communication*, 15(1-2):39–54.

Guoguo Chen, Shuzhou Chai, Guan-Bo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Zhao You, and Zhiyong Yan. 2021. GigaSpeech: An Evolving, Multi-Domain ASR Corpus with 10,000 Hours of Transcribed Audio. In *Interspeech 2021*, pages 3670–3674.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, and 1 others. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505.

Cynthia G Clopper and David B Pisoni. 2004a. Effects of talker variability on perceptual learning of dialects. *Language and speech*, 47(3):207–238.

Cynthia G Clopper and David B Pisoni. 2004b. Some acoustic cues for the perceptual categorization of American English regional dialects. *Journal of Phonetics*, 32(1):111–140.

Michael Cohen, Gay Baldwin, Jared Bernstein, Hy Murveit, and Mitchel Weintraub. 1987. Studies for an adaptive recognition lexicon. *Proc. DARPA Speech Recognition Workshop, Report No. SAIC-87/1644*, pages 71–79.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2023. High fidelity neural audio compression. *Trans. Mach. Learn. Res.*

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*.

John S Garofolo, Lori F Lamel, William M Fisher, David S Pallett, Nancy L Dahlgren, Victor Zue, and Jonathan G Fiscus. 1993. Timit acoustic-phonetic continuous speech corpus. *Linguistic data consortium*.

Mhd Modar Halimeh, Matteo Torcoli, Philipp Grundhuber, and Emanuël AP Habets. 2025. On the relation between speech quality and quantized latent representations of neural codecs. *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, and 1 others. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460.

Mei-Yuh Hwang and Xuedong Huang. 1992. Subphonetic modeling for speech recognition. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Val Jones-Sargent. 1985. *Tyne bytes: a computerised sociolinguistic study of Tyneside*. Peter Lang.

Patricia Keating, B Blankenship, Dani Byrd, Edward Flemming, and Yuichi Todaka. 1992. Phonetic analyses of the TIMIT corpus of American English. *Proc. ICSLP 1992*, pages 823–826.

Patricia A Keating, Dani Byrd, Edward Flemming, and Yuichi Todaka. 1994. Phonetic analyses of word and segment variation using the TIMIT corpus of American English. *Speech Communication*, 14(2):131–142.

Tom Labiausse, Laurent Mazaré, Edouard Grave, Alexandre Défossez, and Neil Zeghidour. 2025. High-fidelity simultaneous speech-to-speech translation. In *International Conference on Machine Learning*, pages 32116–32129. PMLR.

Ineke Mennen and Areti Okalidou. 2006. Acquisition of greek phonology: an overview. *QMU Speech Science Research Centre Working Papers*, (WP-11).

David R Miller and James Trischitta. 1996. Statistical dialect classification based on mean phonetic features. 4:2025–2027.

Pooneh Mousavi, Gallil Maimon, Adel Moumen, Darius Petermann, Jiatong Shi, Haibin Wu, Haici Yang, Anastasia Kuznetsova, Artem Ploujnikov, Ricard Marxer, and 1 others. 2025. Discrete audio tokens: More than a survey! *Transactions on Machine Learning Research*.

Lawrence R Rabiner. 2002. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Samir Sadok, Julien Hauret, and Éric Bavu. 2025. Bringing Interpretability to Neural Audio Codecs. In *Interspeech 2025*, pages 5023–5027.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2022. Utmos: Utokyo-sarulab system for voicemos challenge 2022. *Proceedings of the Annual Conference of the International Speech Communication Association, Interspeech*, 2022:4521–4525.

Thomas Schatz, Vijayaditya Peddinti, Francis Bach, Aren Jansen, Hynek Hermansky, and Emmanuel Dupoux. 2013. Evaluating speech features with the minimal-pair ABX task: analysis of the classical MFC/PLP pipeline. pages 1781–1785.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 2021a. Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. 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 993–1003.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, and Juan and Pino. 2021b. VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. *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 993–1003.

Liming Wang, Mark Hasegawa-Johnson, and Chang Yoo. 2023. A theory of unsupervised speech recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1192–1215.

Zhen Ye, Peiwen Sun, Jiahe Lei, Hongzhan Lin, Xu Tan, Zheqi Dai, Qiuqiang Kong, Jianyi Chen, Jiahao Pan, Qifeng Liu, and 1 others. 2025. Codec does matter: Exploring the semantic shortcoming of codec for audio language model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(24):25697–25705.

Steve Young and Philip Woodland. 1994. State clustering in hidden markov model-based continuous speech recognition. *Computer Speech & Language*, 8(4):369–383.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. 2024. SpeechTokenizer: Unified Speech Tokenizer for Speech Large Language Models. *The Twelfth International Conference on Learning Representations*.

Misalignments in Common Ground as a Bridge Between Pragmatic Theory and LLM Evaluation

Judith Sieker and Sina Zarrieß
Computational Linguistics, Department of Linguistics
Bielefeld University, Germany
{j.sieker;sina.zarriess}@uni-bielefeld.de

Abstract

In this position paper, we argue that misalignments in common ground are not marginal failures of communication, but central diagnostic moments for pragmatic competence, and should therefore play a key role in the evaluation of Large Language Models (LLMs). Evaluating how models respond to such instances of mismatched or incomplete understanding moves beyond surface fluency and correctness, targeting pragmatic competence at a deeper, interactional level. At the same time, misalignments provide controlled settings for testing linguistic theories of common ground, repair, or accommodation – areas that are often difficult to investigate in human communication. We argue that this dual role makes misalignments a natural bridge between pragmatic theory and LLM evaluation.

1 Introduction

At the heart of successful communication lies the concept of common ground – the shared knowledge, beliefs, and assumptions that speakers rely on to understand one another. This shared foundation allows conversations to flow smoothly, with speakers and listeners working together to maintain mutual understanding (Clark, 1996). Yet, in everyday communication, achieving and maintaining common ground is rarely flawless. Time constraints and the need to prioritize information often lead speakers to make assumptions about their audience's knowledge (Beaver, 1999), which can result in misalignments in common ground (Elder and Beaver, 2022). Such misalignments are not rare exceptions but a routine part of communication. For example, imagine the scenario where a local gives directions to a tourist. The local might say, "Turn right after where Miller's bakery used to be," presupposing that the listener knows this bakery and where it was located. A tourist unfamiliar with the area lacks this shared knowledge, resulting in a misalignment in common ground. These moments signal where shared assumptions fail and must be actively negotiated and resolved in order for the interaction to progress, requiring pragmatic strategies such as repair or accommodation. Without resolving misalignments, conversations risk stalling, misunderstandings, or the spread of false information. In this position paper, we argue that misalignments in common ground offer a valuable lens for studying communicative competence, not despite, but because they introduce challenges requiring pragmatic reasoning. They provide a window into whether a language user can detect when shared understanding has broken down and take appropriate steps to restore it.

As Large Language Models (LLMs) are now widely deployed in everyday applications, it becomes increasingly important to assess their ability not just to produce fluent and informative text, but also their capacity to engage in context-sensitive dialogue that is robust to misunderstanding. It remains largely an open question to what extent they can detect and respond to situations where assumptions misalign, expectations are violated, or clarification is needed. By examining LLM behavior in such cases, we gain insight into how they manage the very conditions that define real-world communication. At the same time, modeling and testing how humans resolve such misalignments remains challenging, creating an opportunity for LLMs to serve as controlled testbeds for investigating theories of common ground and communicative grounding. In other words, LLMs can serve not only as objects of evaluation, but also as tools for probing pragmatic theories under controlled conditions. We therefore argue that misalignments in common ground provide a natural bridge between formal pragmatics and computational evaluation, allowing insights from linguistic theory and model behavior to inform one another.

Third Workshop on the Bridges and Gaps between Formal and Computational Linguistics, pages 74–81

2 Misalignments in Common Ground

The concept of common ground has been highly influential in the study of dialogue (Bender and Lascarides, 2019) and plays a central role in many areas of pragmatics, including reference, presupposition, implicature, speech acts, language conventions, and fiction (Geurts, 2024). However, common ground does not arise automatically in conversation. Rather, speakers and listeners work collaboratively to establish and maintain it, a process known as communicative grounding (Larsson, 2018; Chandu et al., 2021). Communicative grounding involves ensuring that participants accurately perceive, understand, and accept each other's utterances (Chandu et al., 2021). To accomplish this, speakers commonly engage in dialogue acts such as asking for clarification or posing follow-up questions (Shaikh et al., 2024). While this process can be straightforward, it often requires addressing and resolving misalignments in the common ground (Larsson, 2018).

Misalignments in common ground refer to instances of mismatched or incomplete understanding between participants in a conversation. Unlike simple misunderstandings, misalignments specifically involve a gap or mismatch in what interlocutors assume to be shared knowledge. They can be overt – recognized and possibly acknowledged – or covert – unrecognized and therefore more difficult to resolve. Furthermore, misalignments in common ground can take different forms, including inconsistencies in what interlocutors believe to be shared knowledge, or gaps that reflect missing or incomplete information (Elder and Beaver, 2022). For example, a speaker referencing "last week's meeting" may misassume that the listener was present. Or a local giving directions may presuppose knowledge the tourist lacks. In both cases, successful communication depends on resolving the misalignment, either implicitly through accommodation, where the listener silently adjusts their knowledge to align with the speaker's presuppositions (von Fintel, 2008; Beaver et al., 2024), or explicitly through repair, such as asking for clarification or correcting the assumption (Clift, 2014; Birkner et al., 2020). Such everyday cases highlight that misalignments are a routine consequence of the fact that no speaker has exhaustive knowledge of the world; assumptions frequently fail to align perfectly with a listener's perspective.

A central question is what conditions influence whether conversation participants choose to accommodate or repair misalignments. Existing research has explored, for example, the social incentives behind repair, highlighting it as an explicit acknowledgment of communication problems (Robinson, 2014), or as a socially motivated strategy (Bernstein, 2016; Elder and Beaver, 2022). However, modeling when and how speakers respond to misalignments remains challenging. While theoretical frameworks provide first insights (Elder and Beaver, 2022), empirical and cognitive research suggests that speakers do not always behave fully collaboratively, complicating attempts to model their behavior in a unified way. For instance, speakers may act egocentrically (Horton and Keysar, 1996; Keysar, 2007), overestimate alignment with their interlocutor (Keysar and Henly, 2002), or have only partial awareness of the common ground (Brown-Schmidt, 2012). Compounding this, not every apparent misalignment reflects a genuine breakdown: hearers sometimes interpret unfamiliar references as informative rather than presupposition-failing, effectively accommodating the new information rather than triggering explicit repair (Heller et al., 2012). As a result, even for human communication, misalignments are difficult to investigate systematically.

3 LLMs and the Challenge of Grounding

To address this challenge, LLMs may offer a useful testbed: they allow misalignments to be studied under controlled conditions and can thus serve as tools for probing theories of accommodation and repair. More broadly, recent work has argued that LLMs can function as testbeds for evaluating linguistic and psychological hypotheses under controlled conditions (Contreras Kallens et al., 2023; Demszky et al., 2023; Millière, 2024). For instance, their scale enables testing across a wide range of contexts, and their intervenability allows specific aspects of the input or model behavior to be manipulated. At the same time, LLMs face key limitations: Handling pragmatic phenomena depends on world knowledge and social norms, and LLMs differ from humans in important ways, for example in their training conditions and lack of social interaction (Hu, 2023; Mahowald et al., 2023; Millière, 2024). This highlights the need for such considerations when using LLMs to study linguistic phenomena and theories, including those related to (misalignments in) common ground.

Beyond their role as experimental tools, LLMs are themselves participants in real-world communicative settings, making it essential to understand their pragmatic behavior in practice. These models have demonstrated strong performance on tasks like summarization and question answering (Millière, 2024), and user-facing tools such as ChatGPT (Achiam et al., 2023) have enabled widespread public experimentation. At the same time, LLMs can remain sensitive to surface-level input variation and may generalize inconsistently across linguistic contexts, pointing to limitations in their language use (Chang and Bergen, 2024). This has motivated a growing body of linguistically-informed research probing the underlying knowledge and behavior of LLMs. Much of this work focuses on syntactic abilities, where LLMs often perform near-human-level (Hu et al., 2020; Schuster and Linzen, 2022; Mahowald et al., 2023). Other work has extended this line of inquiry to semantic and pragmatic phenomena. Here, earlier studies frequently identified clear shortcomings: models faced limitations when interpreting sarcasm (Hu et al., 2023), understanding implicature and presuppositions (Cong, 2022; Ruis et al., 2022; Fried et al., 2023; Sieker and Zarrieß, 2023), handling implicit causality (Zarrieß et al., 2022; Sieker et al., 2023), or responding appropriately to theory-of-mind reasoning (Sap et al., 2022; Shapira et al., 2023; Trott et al., 2023; Gandhi et al., 2024)), suggesting weaknesses at the discourse level of language production (Chang and Bergen, 2024; Contreras Kallens et al., 2023). More recent studies, however, paint a more nuanced picture: especially larger and more recent LLMs can exhibit improved performance on certain pragmatic phenomena, including, for instance, selected aspects of implicature (Cong, 2024; Askari et al., 2025), implicit causality (Kankowski et al., 2025a,b), elicitur (Pan and Kehler, 2025), or politeness (Andersson and McIntyre, 2025).

Crucially, despite these advances, systematic evaluations of how LLMs engage in conversational grounding remain scarce. Most work has focused on assessing and improving their static knowledge (Fierro et al., 2024; Xu et al., 2024), largely overlooking the fact that effective communication requires aligning knowledge dynamically with an interlocutor (Larsson, 2018). In particular, systematic studies of communicative grounding in instruction-tuned LLMs like GPT-4 are only just beginning to emerge. Earlier work has qualitatively examined grounding failures in pretrained models, highlighting their limitations in establishing and maintaining shared understanding (Fried et al., 2023). Benotti and Blackburn (2021), for example, find that systems such as BlenderBot frequently fail to repair misunderstandings or establish shared meaning, suggesting a lack of mechanisms for collaborative grounding. Findings by Lee et al. (2024) and Shaikh et al. (2024) further support the observation that LLMs tend to assume shared understanding rather than actively negotiating it, likely reflecting their optimization for prompt following rather than for the nuanced and collaborative coordination that grounding requires. This pattern is consistent with research on LLM sycophancy, which shows that models often prioritize user approval over factual accuracy, aligning responses with user beliefs even when these are false (Perez et al., 2023; Rrv et al., 2024). More broadly, Chandu et al. (2021) observe that NLP research frequently overlooks key aspects of grounding emphasized in Cognitive Science, such as coordination and mutual understanding, revealing a conceptual gap that may underlie these persistent shortcomings.

Although these studies demonstrate initial progress in operationalizing grounding, relevant tasks and systematic benchmarks for evaluating common ground are still limited. One particularly overlooked aspect is how LLMs resolve misalignments in common ground – a crucial component of effective communication that, to date, has not been systematically investigated. A notable exception is Sarkar et al. (2025), who study misalignments in goal-oriented Ubuntu chat logs and show that LLMs particularly struggle to detect implicit ones.

Handling such cases requires more than grammaticality or lexical choice: it involves interpreting intentions, recognizing when communication has broken down, and deciding whether to accommodate or repair. This, in turn, demands consideration of the broader communicative context – something that has largely been absent from existing probing studies. This gap is especially concerning given the increasing use of LLMs in sensitive settings like healthcare or legal assistance, where pragmatic failures can have serious consequences. For instance, if LLMs fail to resolve misalignments in common ground but still respond fluently and confidently, they risk reinforcing user misconceptions (Fried et al., 2023). Investigating how LLMs manage common ground and repair misalignments is therefore not only a theoretical concern but also crucial for their responsible deployment.

4 Misalignments as a Lens for Evaluation

Misalignments in common ground are not marginal anomalies in communication; they are a natural, frequent, and revealing part of interaction. We argue that such misalignments should not be treated as noise or edge cases in LLM evaluation, but as core phenomena for assessing pragmatic competence. Furthermore, we argue that when investigating misalignments in common ground, we can assign LLMs a dual role. On the one hand, they are objects of evaluation: studying how LLMs respond to misalignments provides a direct probe of their pragmatic competence. On the other hand, they can serve as tools for investigation, as their controllability enables the construction of experimental settings in which assumptions about shared knowledge can be systematically varied – something that is difficult to achieve in studies of human interaction. In this way, misalignments offer a bridge between formal and computational approaches, linking theoretical accounts of common ground to empirical evaluation.

One way to instantiate such an evaluation paradigm is through linguistically well-defined phenomena. As an illustrative example, consider presuppositions. Presuppositions are implicit assumptions that speakers take for granted as shared knowledge (Stalnaker, 1973). They are essential for understanding how speakers rely on shared background knowledge, without making it explicit (Schwarz, 2019). For instance, the question "Has your sister finished reading the book I lent her?" conveys multiple presuppositions: that the listener has a sister, that the speaker lent her a book, and that she started reading it. One intriguing aspect of presuppositions, central to the study of misalignments in common ground, is *presupposition failure*, which occurs when the assumed background knowledge is actually false (Yablo, 2006). For instance, if the listener has no sister, is not aware of their sister borrowing the book, or if she has not actually started reading it, a presupposition fails, potentially causing a misalignment in common ground.

Recent work has shown that such false presuppositions can be used to probe LLMs' pragmatic competence in handling misalignments in common ground. For instance, Sieker et al. (2025) and Lachenmaier et al. (2025) systematically embedded false presuppositions in prompts, analyzing whether LLMs were able to reject the implied assumptions. With their evaluation framework, the authors show that false presuppositions are especially useful for the evaluation of misalignments in common ground: while the misalignment introduced by presuppositions is triggered implicitly, it requires an explicit intervention. Importantly, beyond demonstrating limitations in models' ability to handle such cases, the authors also show that linguistically controlled factors – such as the type of presupposition trigger or contextual framing – systematically affect model behavior. This allows model responses to inform not only assessments of pragmatic competence, but also questions central to linguistic theories of presupposition.

More broadly, linguistic phenomena such as presuppositions, implicatures (Grice, 1975), or elicitures (Cohen and Kehler, 2021), for example, provide theory-informed, minimal-pair-friendly conditions that can serve as evaluation paradigms to systematically probe misalignments in common ground. We encourage the use of such phenomena to develop evaluation frameworks that make misalignments experimentally controllable.

A practical way to operationalize this approach is to construct minimal pairs that systematically manipulate whether a misalignment is present. To illustrate this with a different phenomenon, consider implicature.

A minimal pair could consist of two contexts that are identical except for one piece of shared information. In context A (no misalignment), the prior discourse is neutral. A teacher states, "I graded the exams", after which the utterance "Some of the students passed the exam" licenses the standard implicature that not all students passed. In context B (misalignment), the prior discourse establishes that all students passed the exam. For instance, the teacher states, "Everyone did really well this time", after which the same utterance "Some of the students passed" creates a mismatch between the implicature and the shared contextual information.

In both contexts, a model is then presented with a neutral follow-up question, and its response can be scored on whether it ignores the tension and accommodates silently, or explicitly repairs it, for instance, by flagging the misalignment before answering. A key advantage of using LLMs in this paradigm is their controllability at multiple levels (Demszky et al., 2023). At the level of the prompt, linguistic factors can be systematically varied. For instance, scalar items ("some" vs. "many" vs. "most") can be substituted to test whether im-

plicature strength affects model behavior, or discourse contexts can be varied to examine how different settings affect the detection and resolution of misalignments. At the level of the model, (open-source) LLMs allow for intervention in training data, model size, or fine-tuning. In this way, LLMs can serve as a controlled testbed for investigating how implicatures are derived and interpreted and, more generally, how misalignments in common ground are detected and resolved.

5 Conclusion

In this position paper, we have argued that misalignments in common ground are not exceptions to effective communication but central to its structure, and that they create a concrete setting in which linguistic theory and LLM-based experimentation can productively inform one another. By examining how LLMs behave in situations where shared assumptions fail, we gain a twofold perspective: instances of misalignment provide theory-informed diagnostics of pragmatic competence, while LLMs offer controlled testbeds for investigating questions central to linguistic theory. We therefore invite the research community to place greater emphasis on the diagnostic moments of misalignments in common ground, and to develop tools and benchmarks that make communicative grounding and misalignment resolution a core part of model evaluation.

Limitations

There are limitations to this paper.

First, we focus on presuppositions and implicatures as illustrative examples because they are linguistically well-defined, amenable to minimal-pair designs, and supported by established experimental traditions. Misalignments in common ground, however, can arise through a much broader range of pragmatic phenomena. Future work should therefore investigate how the proposed framework extends to additional sources of misalignment.

Second, our example evaluation paradigm focuses on single-turn interactions, which allow for controlled, minimal-pair-based evaluation but cannot fully capture the dynamic, multi-turn negotiation of meaning that characterizes communicative grounding (Clark, 1996). Such paradigms nevertheless provide a useful starting point, offering a level of experimental control that is difficult to achieve in fully interactive settings. Exploring how controlled pragmatic evaluation can be extended to multi-turn interactions represents an important direction for future work. Recent work by Lachenmaier et al. (2026), for instance, demonstrates how linguistically grounded notions of repair can be operationalized in a multi-turn setting, pointing toward promising directions for such extensions.

Third, we do not address broader pragmatic capacities such as Theory of Mind (Holterman and van Deemter, 2023), which are closely related to common-ground reasoning but represent a distinct dimension of communicative competence. Understanding how such abilities interact with the detection and resolution of misalignments remains an important direction for future research.

Acknowledgements

We want to thank the anonymous reviewers for their very helpful comments and suggestions.

We acknowledge financial support from the project “SAIL: SustAInable Life-cycle of Intelligent Socio-Technical Systems" (Grant ID NW21-059A), an initiative of the Ministry of Culture and Science of the State of North Rhine-Westphalia.

References

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

Marta Andersson and Dan McIntyre. 2025. Can chatgpt recognize impoliteness? an exploratory study of the pragmatic awareness of a large language model. *Journal of Pragmatics*, 239:16–36.

Raha Askari, Sina Zarrieß, Özge Alacam, and Judith Sieker. 2025. Are BabyLMs deaf to Gricean maxims? a pragmatic evaluation of sample-efficient language models. In *Proceedings of the First BabyLM Workshop*, pages 52–65, Suzhou, China. Association for Computational Linguistics.

D Beaver. 1999. Presupposition accommodation: a plea for common sense. pages 21–44.

David I. Beaver, Bart Geurts, and Kristie Denlinger. 2024. Presupposition. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*, Fall 2024 edition. Metaphysics Research Lab, Stanford University.

Emily M Bender and Alex Lascarides. 2019. Linguistic fundamentals for natural language processing II: 100 essentials from semantics and pragmatics. *Synth. Lect. Hum. Lang. Technol.*, 12(3):1–268.

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.

Katie A Bernstein. 2016. "misunderstanding" and (mis)interpretation as strategic tools in intercultural interactions between preschool children. *Appl. Linguist. Rev.*, 7(4):471–493.

Karin Birkner, Peter Auer, Angelika Bauer, and Helga Kotthoff. 2020. *Einführung in Die Konversationsanalyse*. de Gruyter Studium. De Gruyter, Berlin, Germany.

Sarah Brown-Schmidt. 2012. Beyond common and privileged: Gradient representations of common ground in real-time language use. *Language and Cognitive Processes*, 27(1):62–89.

Khyathi Raghavi Chandu, Yonatan Bisk, and Alan W Black. 2021. Grounding 'grounding' in NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4283–4305, Online. Association for Computational Linguistics.

Tyler A. Chang and Benjamin K. Bergen. 2024. Language model behavior: A comprehensive survey. *Computational Linguistics*, 50(1):293–350.

Herbert H. Clark. 1996. *Using Language*. "Using" Linguistic Books. Cambridge University Press.

Rebecca Clift. 2014. 4. conversation analysis. In *Pragmatics of Discourse*, pages 97–124. DE GRUYTER, Berlin, München, Boston.

Jonathan Cohen and Andrew Kehler. 2021. Conversational eliciture. *Philosophers' Imprint*, 21(12).

Yan Cong. 2022. Psycholinguistic diagnosis of language models' commonsense reasoning. In *Proceedings of the First Workshop on Commonsense Representation and Reasoning (CSRR 2022)*, pages 17–22, Dublin, Ireland. Association for Computational Linguistics.

Yan Cong. 2024. Manner implicatures in large language models. *Scientific Reports*, 14(1):29113.

Pablo Contreras Kallens, Ross Deans Kristensen-McLachlan, and Morten H Christiansen. 2023. Large language models demonstrate the potential of statistical learning in language. *Cogn. Sci.*, 47(3):e13256.

Dorottya Demszky, Diyi Yang, David S. Yeager, Christopher J. Bryan, Margarett Clapper, Susannah Chandhok, Johannes C. Eichstaedt, Cameron Hecht, Jeremy Jamieson, Meghann Johnson, Michaela Jones, Danielle Krettek-Cobb, Leslie Lai, Nirel JonesMitchell, Desmond C. Ong, Carol S. Dweck, James J. Gross, and James W. Pennebaker. 2023. Using large language models in psychology. *Nature Reviews Psychology*, 2(11):688–701.

Chi-He Elder and David Beaver. 2022. "we're running out of fuel!": When does miscommunication go unrepaired? *Intercult. Pragmat.*, 19(5):541–570.

Constanza Fierro, Ruchira Dhar, Filippos Stamatiou, Nicolas Garneau, and Anders Søgaard. 2024. Defining knowledge: Bridging epistemology and large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16096–16111, Miami, Florida, USA. Association for Computational Linguistics.

Daniel Fried, Nicholas Tomlin, Jennifer Hu, Roma Patel, and Aida Nematzadeh. 2023. Pragmatics in language grounding: Phenomena, tasks, and modeling approaches. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12619–12640, Singapore. Association for Computational Linguistics.

Kanishk Gandhi, J.-Philipp Fränken, Tobias Gerstenberg, and Noah D. Goodman. 2024. Understanding social reasoning in language models with language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Bart Geurts. 2024. Common Ground in Pragmatics. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*, Winter 2024 edition. Metaphysics Research Lab, Stanford University.

Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Daphna Heller, Kristen S. Gorman, and Michael K. Tanenhaus. 2012. To name or to describe: Shared knowledge affects referential form. *Topics in Cognitive Science*, 4(2):290–305.

Bart Holterman and Kees van Deemter. 2023. Does chatgpt have theory of mind? *Preprint*, arXiv:2305.14020.

William S. Horton and Boaz Keysar. 1996. When do speakers take into account common ground? *Cognition*, 59(1):91–117.

Jennifer Hu. 2023. *Neural language models and human linguistic knowledge*. Ph.D. thesis, Massachusetts Institute of Technology.

Jennifer Hu, Sammy Floyd, Olessia Jouravlev, Evelina Fedorenko, and Edward Gibson. 2023. A fine-grained comparison of pragmatic language understanding in humans and language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4194–4213, Toronto, Canada. Association for Computational Linguistics.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment

of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Florian Kankowski, Torgrim Solstad, Sina Zarriess, and Oliver Bott. 2025a. Implicit causality-biases in humans and llms as a tool for benchmarking llm discourse capabilities. *Preprint*, arXiv:2501.12980.

Florian Kankowski, Torgrim Solstad, Sina Zarrieß, and Oliver Bott. 2025b. Instruction tuning modulates discourse biases in language models. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 47, pages 2818–2826.

Boaz Keysar. 2007. Communication and miscommunication: The role of egocentric processes. *Intercultural Pragmatics*, 4(1).

Boaz Keysar and Anne S Henly. 2002. Speakers' overestimation of their effectiveness. *Psychological Science*, 13(3):207–212.

Clara Lachenmaier, Hannah Bultmann, and Sina Zarrieß. 2026. Talking to a know-it-all gpt or a second-guesser claude? how repair reveals unreliable multi-turn behavior in llms. *Preprint*, arXiv:2604.19245.

Clara Lachenmaier, Judith Sieker, and Sina Zarrieß. 2025. Can LLMs ground when they (don't) know: A study on direct and loaded political questions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14956–14975, Vienna, Austria. Association for Computational Linguistics.

Staffan Larsson. 2018. Grounding as a side-effect of grounding. *Top. Cogn. Sci.*, 10(2):389–408.

Hyunji Lee, Se June Joo, Chaeeun Kim, Joel Jang, Doyoung Kim, Kyoung-Woon On, and Minjoon Seo. 2024. How well do large language models truly ground? 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 2437–2465, Mexico City, Mexico. Association for Computational Linguistics.

Kyle Mahowald, Anna A Ivanova, Idan A Blank, Nancy Kanwisher, Joshua B Tenenbaum, and Evelina Fedorenko. 2023. Dissociating language and thought in large language models.

Raphaël Millière. 2024. Language models as models of language. *Preprint*, arXiv:2408.07144.

Dingyi Pan and Andrew Kehler. 2025. Pragmatic competence in LLMs: The case of eliciture. In *Proceedings of the Society for Computation in Linguistics 2025*, pages 388–391, Eugene, Oregon. Association for Computational Linguistics.

Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Benjamin Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, and 44 others. 2023. Discovering language model behaviors with model-written evaluations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13387–13434, Toronto, Canada. Association for Computational Linguistics.

Jeffrey D. Robinson. 2014. What "what?" 'tells us about how conversationalists manage intersubjectivity. *Research on Language and Social Interaction*, 47(2):109–129.

Aswin Rrv, Nemika Tyagi, Md Nayem Uddin, Neeraj Varshney, and Chitta Baral. 2024. Chaos with keywords: Exposing large language models sycophancy to misleading keywords and evaluating defense strategies. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12717–12733, Bangkok, Thailand. Association for Computational Linguistics.

Laura Ruis, Akbir Khan, Stella Biderman, Sara Hooker, Tim Rocktäschel, and Edward Grefenstette. 2022. Large language models are not zero-shot communicators. *arXiv [cs.CL]*.

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.

Rupak Sarkar, Neha Srikanth, Taylor Pellegrin, Rachel Rudinger, Claire Bonial, and Philip Resnik. 2025. Understanding common ground misalignment in goal-oriented dialog: A case-study with Ubuntu chat logs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3200–3215, Vienna, Austria. Association for Computational Linguistics.

Sebastian Schuster and Tal Linzen. 2022. When a sentence does not introduce a discourse entity, transformer-based models still sometimes refer to it. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 969–982, Seattle, United States. Association for Computational Linguistics.

Florian Schwarz. 2019. *Presuppositions, Projection, and Accommodation*. Oxford University Press.

Omar Shaikh, Kristina Gligoric, Ashna Khetan, Matthias Gerstgrasser, Diyi Yang, and Dan Jurafsky. 2024. Grounding gaps in language model generations. 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 6279–6296, Mexico City, Mexico. Association for Computational Linguistics.

Natalie Shapira, Guy Zwirn, and Yoav Goldberg. 2023. How well do large language models perform on faux pas tests? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10438–10451, Toronto, Canada. Association for Computational Linguistics.

Judith Sieker, Oliver Bott, Torgrim Solstad, and Sina Zarrieß. 2023. Beyond the bias: Unveiling the quality of implicit causality prompt continuations in language models. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 206–220, Prague, Czechia. Association for Computational Linguistics.

Judith Sieker, Clara Lachenmaier, and Sina Zarrieß. 2025. LLMs struggle to reject false presuppositions when misinformation stakes are high. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 47.

Judith Sieker and Sina Zarrieß. 2023. When your language model cannot Even do determiners right: Probing for anti-presuppositions and the maximize presupposition! principle. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 180–198, Singapore. Association for Computational Linguistics.

Robert Stalnaker. 1973. Presuppositions. *Journal of Philosophical Logic*, 2(4):447–457.

Sean Trott, Cameron Jones, Tyler Chang, James Michaelov, and Benjamin Bergen. 2023. Do large language models know what humans know? *Cognitive Science*, 47(7):e13309.

Kai von Fintel. 2008. What is presupposition accommodation, again? *Philosophical Perspectives*, 22(1):137–170.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024. Knowledge conflicts for LLMs: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8541–8565, Miami, Florida, USA. Association for Computational Linguistics.

S Yablo. 2006. Non-catastrophic presupposition failure.

Sina Zarrieß, Hannes Groener, Torgrim Solstad, and Oliver Bott. 2022. This isn't the bias you're looking for: Implicit causality, names and gender in German language models. In *Proceedings of the 18th Conference on Natural Language Processing (KONVENS 2022)*, pages 129–134, Potsdam, Germany. KONVENS 2022 Organizers.

Towards Benchmarking Old Church Slavonic Lemmatization

Usman Nawaz[1] **Marianna Napolitano**[2] **Iris Karafillidis**[3]
Liliana Lo Presti[1] **Marco La Cascia**[1]

[1]Department of Engineering, University of Palermo, Palermo, Italy
[2]University of Modena and Reggio Emilia / FSCIRE, Italy
[3]University of Padua, Padua, Italy
mariannanapolitano@unimore.it

Abstract

Lemmatization is an important preprocessing step in Natural Language Processing (NLP); however, annotated resources for medieval languages such as Old Church Slavonic (OCS) are limited in scope, size, and diversity. This paper presents the annotated resources for OCS lemmatization, including annotation process, design choices and non-standard Unicode related issues. The annotated corpus is used to evaluate existing lemmatization tools (Stanza and UDPipe-2 models trained on the UD 2.12 treebank, and a dictionary-based approach) both in cross-dataset and on a corpus obtained by merging the new annotations with existing UD V2.12 OCS data. Pretrained models perform poorly ($\approx$ 15–16%), below a dictionary baseline ($\approx$ 38%), while retraining on the new data improves performance (up to $\approx$ 51%) and shows different cross-dataset generalization. Experiments in cross-dataset and on the combined corpus demonstrate that lemmatization performance depends strongly on dataset similarity, annotation conventions, and orthographic mismatch. Overall, the findings show the value of the newly annotated resources and the importance of extending OCS lemmatization benchmarks for historical Slavic NLP.

1 Introduction

OCS holds a unique place in Slavic linguistic history, diachrony, and manuscript variation (Picchio, 1972; Garzaniti, 2019; Naumow, 1983; Zhivov, 2009) and plays an important role in the study of early Slavic written culture and the evolution of the Cyrillic and Glagolitic scripts (Picchio, 1972; Garzaniti, 2019; Hetényi and Ivanič, 2021).

In recent times, OCS texts have become more prominent in computational linguistics with the development of treebanks, corpora, and other applications of NLP (Eckhoff and Berdicevskis, 2015). Although several annotated OCS resources are available, benchmark-quality lemmatization datasets remain limited in size, fragmented across sources and editorial traditions, and difficult to transfer across annotation schemes. Moreover, because OCS is a language associated with the early Slavic literary tradition from the 9th century, expert annotation is substantially harder to scale than for modern languages, where larger annotator pools and standardized resources are more readily available.

Lemmatization transforms words into their base or canonical form, which helps reduce data sparsity and improve downstream tasks, including machine translation (Goldwater and McClosky, 2005; Koehn et al., 2007), information retrieval (Kanis and Skorkovská, 2010; Zeroual and Lakhouaja, 2017), and text analysis for historical texts (Piotrowski, 2012; Manjavacas et al., 2019) and sentiment analysis (Abdul-Mageed et al., 2014). Lemmatization is particularly critical for OCS because the language is characterized by regional and historical diversification and exhibits numerous and significant orthographic and textual variants. Lemmatization can support corpus search, comparison of textual witnesses, analysis of orthographic and morphological variation, and the study of lexical distributions across sources and textual variants. For example, the noun ѩзыкъ, "tongue, language, speech, nation, people", may appear in several orthographic forms, including ꙗзыкъ, ꙗзикь, езыкъ, езикь, езыкь, ѥзыкь, ꙗзыкъ, ꙗзыкъ, ѩз҆ыкъ, ѩзыикь, ѩзыкъ, ѫзыкъ, ѫзыкъ, and ѣзикь, as well as in abbreviated forms such as ѩӟкъ, ѫӟкъ, ꙗӟкъ, and ѣӟкъ. These forms must also be considered across plural and dual number and across the seven cases, as reflected in the Digital OCS Dictionary at GORAZD (SJS, 1966–1997).

Third Workshop on the Bridges and Gaps between Formal and Computational Linguistics, pages 82–93

While modern languages are equipped with standardized datasets (i.e., GLUE (Wang et al., 2018), SuperGLUE (Wang et al., 2019), XGLUE (Liang et al., 2020), etc), historical languages have significantly limited benchmark-quality resources (Dereza et al., 2024). This is particularly applicable to Church Slavonic, which has recently experienced an increase in digital support and tool development, while lemmatization benchmarks remain limited. Among the available resources, PROIEL (Haug and Jøhndal, 2008) introduced treebank-style annotation for early Indo-European texts, including OCS, and TOROT (Eckhoff and Berdicevskis, 2015) for Old Russian and OCS texts. Universal Dependencies (UD) also provides a shared annotation framework for such resources (de Marneffe et al., 2021). Toolkits such as Stanza and UDPipe are also available for UD-style processing and lemmatization (Qi et al., 2020; Straka et al., 2019). More recently, the heterogeneity and fragmentation of Church Slavonic resources have also been highlighted by DIACU (Cassese et al., 2025). However, the availability of digital or annotated resources does not by itself guarantee reliable evaluation on newly prepared OCS texts.

In this work, we describe an ongoing effort in annotating texts for OCS lemma-level evaluation. We present the normalization and encoding steps required for the computational stability of the texts, the annotation process used in the creation of the dataset, and evaluate off-the-shelf systems on the newly annotated datasets. The results show that pretrained systems struggle with the new data, but in-domain retraining leads to significant improvements. At the same time, the results across datasets are low, suggesting that, once again, orthographic, editorial, and annotational differences are a major challenge in OCS lemmatization.

The contributions of this paper are as follows:

- We introduce a newly annotated OCS dataset for lemma-level evaluation comprising 29,678 tokens total, 25,517 unique forms, and 23,803 unique lemmas.
- We document the preprocessing and annotation decisions required to convert heterogeneous OCS materials into a stable evaluation resource, including a normalization workflow for handling source-specific PUA characters and related encoding inconsistencies.
- We benchmark pretrained and retrained models (Stanza, UDPipe-2, and a dictionary baseline) on the new dataset. We compare in-domain and combined-corpus training across test sets. We show that lemmatization performance depends on dataset similarity, orthographic variation, and annotation mismatch, highlighting the difficulty of cross-resource generalization in historical Slavic NLP. Code, model weights, dataset access details, and license information are publicly available on GitHub[1].

The remainder of the paper is structured as follows. Section 2 reviews related work on OCS resources and historical lemmatization. Section 3 describes the source material, annotation scheme, preprocessing decisions, and workflow used to construct the dataset. Section 4 presents the experimental setup, including data splits, systems, and evaluation metrics. Section 5 reports the main benchmarking results and cross-dataset findings. Section 6 discusses the implications of the results, and Section 7 concludes the paper.

2 Related Work

The computational study of OCS has been supported by a limited set of resources including PROIEL for historical Indo-European languages (Haug and Jøhndal, 2008), TOROT for Old Russian (Eckhoff and Berdicevskis, 2015), and the UD annotation framework (de Marneffe et al., 2021). These resources collectively form the foundation for computational analysis of medieval Slavic languages, but it remains unexplored how well they support the generalization of models to unseen datasets.

A recent work aggregates multiple resources in DIACU, a diachronically and geographically annotated Church Slavonic dataset (Cassese et al., 2025). It directly addresses the fragmentation in Church Slavonic resources, which are scattered across corpora, and editorial practices. A similar concern appears in retrieval-based research on Church Slavonic variants, which points to the difficulty of working across chronological and regional variation, as well as heterogeneous source conditions (Lendvai et al., 2025). In other words, the existence of digital text does not automatically imply access to benchmark-quality resources for lemmatization and similar tasks.

[1] https://github.com/usmannawaz01/ocs-lemmatization-benchmark

Several methods have been proposed for lemmatization. Rule-based and finite-state approaches are commonly used in settings with substantial expert knowledge and manually constructed resources (Schmid et al., 2004). Dictionary-based methods remain attractive because of their simplicity and usefulness in settings with high lexical overlap (Nawaz et al., 2024). Edit-based approaches have shown strong performance across languages (Straka and Straková, 2017; Straka et al., 2019), while recent neural methods model lemmatization as a character-level transduction problem (Schnober et al., 2016; Kanerva et al., 2021; Bergmanis and Goldwater, 2018). The Stanford lemmatization system (Qi et al., 2018) combines lexicon lookup with neural sequence-to-sequence generation. Lemmas are first retrieved from lexicons built from the training data, and neural generation is used when lookup fails. This approach later became part of the Stanza toolkit (Qi et al., 2020). Stanza and UDPipe are of particular interest with regard to the OCS language, but their performance is correlated with the resources used for training and evaluation, and is sensitive to orthographic variation.

Recent results from the SIGTYP 2024 shared task on historical languages (Dereza et al., 2024) highlight ongoing challenges in evaluation. Even robust approaches such as TartuNLP (Dorkin and Sirts, 2024), Heidelberg-Boston (Riemenschneider and Krahn, 2024), and UDParse (Heinecke, 2024) are affected by scarcity and tokenization differences in historical languages (Dereza et al., 2024). This is particularly relevant for OCS lemmatization, where recently available resources may be used for both training and evaluation.

In light of these challenges, our work focuses not on proposing a new lemmatization architecture, but on resource construction and evaluation. We present a newly annotated OCS dataset for lemma-level benchmarking, describe the normalization and annotation decisions behind it, and use it to assess the portability of existing OCS lemmatization systems.

3 Data Collection and Annotation

The annotated material used in this work includes texts from diverse OCS sources to reflect differences in editorial practices, orthographic systems, and textual genres. The current corpus includes texts from the Cyrillomethodiana[2] portal, creed texts from Pravmir[3], and dictionary texts from Azbuka[4]. Table 1 describes the composition of the annotated sources.

Table 1: OCS source of the annotated texts in the dataset

Source group	Origin	Tokens
Cyrillomethodiana	Cyrillomethodiana portal	5,789
Creed	Pravmir: *Simvol very*	77
Dictionary texts	Azbuka	23,812
Total		29,678

The dataset was created by including texts from sources with varying origins, formats, and textual conventions. This variety supports the evaluation of lemmatization systems under more realistic generalization conditions.

3.1 Normalization and Annotation Policy

The study focuses on lemma annotation at the token level, where each surface form is mapped to its canonical lexical form. This is particularly relevant for OCS due to its extensive inflection and orthographic variations.

Prior to annotation, the source texts were normalized to a consistent Unicode representation. Source-specific Private Use Area (PUA) characters were addressed using manually validated mapping tables (Napolitano and Nawaz, 2025). The purpose of the normalization process was not to eliminate the natural variations that exist in the source texts, but rather to avoid technical characters that become corrupted after scraping and may affect the annotation and evaluation processes. Additional details and representative examples are provided in Appendix A.

Table 2 presents representative examples of PUA characters and their standardized Unicode equivalents: this required systematic character-level remapping rather than simple formatting cleanup.

Table 3 illustrates how raw source text is converted into a stable Unicode representation prior to annotation. PUA characters are resolved into standard Unicode Cyrillic forms, easing downstream tokenization and lemma annotation.

As the next step, lemma annotation was applied to the normalized texts. Where possible,

[2] https://histdict.uni-sofia.bg/textcorpus/list
[3] https://www.pravmir.ru/simvol-very/
[4] https://azbyka.ru/otechnik/Spravochniki/polnyj-pravoslavnyj-tserkovnoslavjanskij-slovar/1

Table 2: Representative examples of source-specific PUA characters and their standardized Unicode equivalents used during normalization.

PUA character	PUA code point	Normalized character	Unicode code point
	U+E205	и	U+0438
~	U+E012	Ѣ	U+0462
	U+E20D	Ꙋ	U+0427
н	U+E201	Ѥ	U+0464
	U+E342	◌͡◌	U+0487
	U+E343	◌҃	U+0488
	U+E20C	Ꙝ	U+041D
⌐	U+E018	ъ	U+044A
	U+E20E	Ѳ	U+042E
	U+E213	Ѥ	U+0464

lemma forms follow existing scholarly conventions and treebank best practices, with contextual and grammatical considerations applied in ambiguous cases, while punctuation and other non-lexical tokens were retained. The present data is restricted to token-level lemmatization and does not introduce a separate multi-word expression annotation layer. This follows UD OCS tokenization, where there are no multi-word tokens[5].

Table 3: Example of text before and after Unicode harmonization.

Stage	Text snippet
Before normalization (Source Text)	Странꙗнолюбⷭемь цѣлꙋ ѣкоже древле Авраамъ· вꙑшьнѣго доⷣе свѣтла селенⷭѣ· въ немьже блаженѥ прѣдъстоа· сⷭѣшⷭ ѣко ⷭ слъньце· Тѣмьже ⷭ насъ просвѣтⷭ нꙑнѣ ⷭьтжщⷭхъ прѣсвѣтлоѥ твоѥ трьжьство
After normalization (PUA characters remapping)	Странꙗнолюбиемь цѣлꙋ ѣкоже древле Авраамъ· вꙑшьнѣго доиде свѣтла селениѣ· въ немьже блаженѥ прѣдъстоа· сиѣеши ѣко и слъньце· Тѣмьже и насъ просвѣти нꙑнѣ ѣьтжщиихъ прѣсвѣтлоѥ твоѥ трьжьство:

3.2 Annotation Workflow

The annotation workflow includes preprocessing, candidate organization, expert annotation, and multiple-stage verification. The data collection was performed by scraping the relevant websites. During preprocessing, texts are cleaned and normalized to handle PUA characters and encoding issues. This process results in a consistent text representation suitable for tokenization and annotation.

Annotation was conducted semi-automatically by taking advantage of a lookup lexicon built by merging lemma entries from UD 2.12 OCS with the lemmas manually annotated and iteratively obtained through the dataset creation process. To improve efficiency, lemma candidates are grouped based on difficulty by distinguishing between single-match, ambiguous, and unseen forms. In the initial candidate-generation stage for the manually annotated Cyrillomethodiana running-text subset, 1,838 tokens received a single lexicon match, 1,432 tokens received multiple possible lemma candidates, and 2,519 tokens were not covered by the UD OCS lookup lexicon. These correspond to 31.75%, 24.74%, and 43.51% of the 5,789 tokens in this subset, respectively.

This allows for easier prioritization of difficult cases and minimizes redundant work on simpler cases. When a reliable lexicon match was available, the lexicon-based lemma was used as the initial annotation. For tokens not covered by the lexicon, the lemma prediction produced by the pretrained Stanza model (Version 1.5.1) for OCS was used. The initial lemma annotations were manually verified and, where necessary, corrected and reviewed by an expert linguist, resulting in a dataset verified through expert verifications.

The overall annotation workflow in Figure 1 produces human-verified lemma annotations that can be extended over time as more texts are included. Although the present work targets lemma annotation only, the corpus is already organized in CoNLL-U format (Straka et al., 2016) to ensure compatibility with the UD framework and to support future extensions. Whenever possible, sentence boundaries are also preserved from the source material, which facilitates future expansion toward fuller UD-style annotation (Straka et al., 2016).

3.3 Dataset Statistics

The current annotated dataset consists of 29,678 tokens drawn from multiple OCS source groups. In addition to its overall size, the dataset exhibits substantial lexical diversity; it contains 25,517 unique word forms and 23,803 unique lemmas. Beyond the annotated subset used in the present experiments, the broader source collection is substantially larger. It includes more than 200 texts comprising approximately more than 5 million tokens, and may support future annotation efforts.

As shown in Table 4, although the present dataset is smaller than UD 2.12 OCS in total number of tokens, it exhibits substantially greater lemma diversity, as reflected in its much larger number of unique lemmas. Indeed, beyond to-

[5] https://universaldependencies.org/cu/index.html

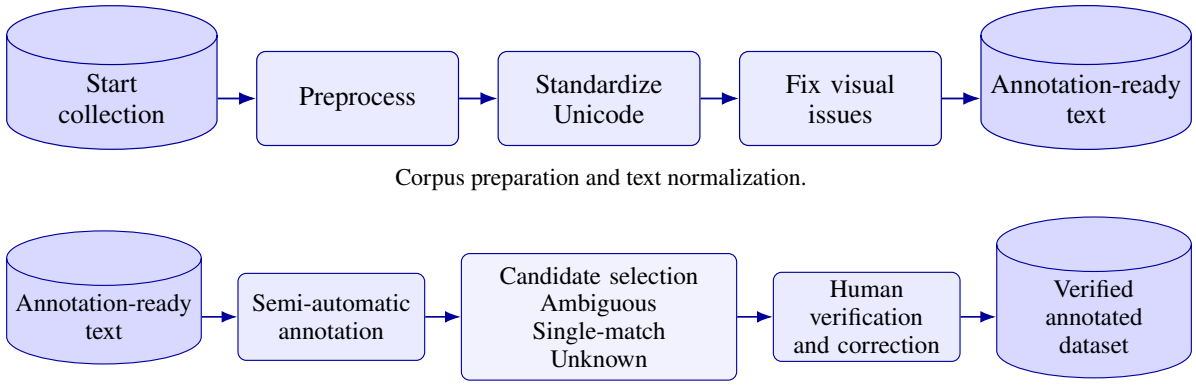


Figure 1: Workflow for corpus preparation and lemma annotation. The upper panel shows data collection and preprocessing, where source-specific Unicode and rendering issues are resolved to produce annotation-ready text. The lower panel shows the annotation stage, moving from semi-automatic annotation through candidate selection and human verification to the final verified annotated dataset.

Table 4: Lexical statistics for our dataset and UD 2.12 OCS. The UD 2.12 OCS values are taken from the official dataset statistics, while the values for our dataset were computed by us.

Dataset	Tokens	Forms	Lemmas
Our	29678	25517	23803
UD 2.12	198843	43815	8247

ken counts, the dataset reflects variation across sources and highlights more challenging generalization conditions. The high ratio of unique forms and lemmas is mainly due to the composition of the current data. In particular, the Azbuka portion consists largely of lexicographical entries rather than continuous running text, which increases lexical diversity and reduces token repetition. Therefore, the dataset should not be interpreted as a frequency-balanced sample of natural OCS. Instead, it provides a challenging lemma-level evaluation set with broad lexical coverage and many low-frequency forms.

4 Experimental Protocol and Baseline Methods

The annotated OCS material is partitioned into training, development, and test sets using an approximate 80/10/10 split at the sentence level, preserving sentence boundaries within each source file. This results in 23,768 training tokens, 2,979 development tokens, and 2,931 test tokens (Table 5).

Table 5: Train/dev/test split of the annotated OCS material.

Split	Tokens
Train	23768
Dev	2979
Test	2931
Total	29678

Following common practice in UD-based and historical NLP (Dereza et al., 2024), the training and development splits are used for model training and selection, while the test split is reserved for evaluation. Additionally, we propose cross-dataset evaluation on existing OCS resources to assess generalization under variation in orthography and editorial conventions.

In the experiments presented in this paper, four setups for lemmatization of OCS texts are compared. First, a pre-trained Stanza lemmatizer for OCS based on the UD 2.12 treebank (Qi et al., 2020). The Stanza lemmatizer uses lexicon-based memorization in conjunction with a neural character-level seq2seq model in its backoff stage. Second, a pre-trained UDPipe 2 model trained on the same treebank release (Straka and Straková, 2017; Straka et al., 2019), uses an edit-tree classification approach for lemmatization, predicting a word-to-lemma transformation rule. Third, a dictionary-based baseline constructed from the training split. For any surface word form in the test set, it predicts the most frequently associated lemma seen in this training data; for any unseen word form, it predicts the unchanged surface form itself. Finally, two retrained Stanza settings are evaluated. In the first setting, the Stanza pipeline is trained on the annotated OCS training split only. In the second setting, the Stanza pipeline is trained

on a combined corpus formed by merging the existing UD 2.12 OCS data with the new annotations. In both settings, the tokenizer and lemmatizer are trained from scratch, the development split is used for model selection, and evaluation is conducted under both gold tokenization and raw-text settings.

Evaluation is performed using the official CoNLL-U 2018 UD script. In the gold-tokenized setting, we report lemma accuracy, and in the raw-text setting, we report lemma F1-score. Beyond overall score, we also evaluate performance on unseen tokens to assess generalization, which is especially important in the context of historical lemmatization. Evaluation is conducted both on the annotated test set and across datasets where appropriate.

5 Results

We evaluate all systems on the annotated test set. Results are reported in Table 6 under both gold tokenization and raw-text settings.

Under gold tokenization, the pre-trained Stanza model achieves 15.56% lemma accuracy, while the pre-trained UDPipe 2 model achieves 16.27%, indicating that both pre-trained systems generalize poorly on the annotated test set.

The best-performing model is the Stanza model trained on the annotated data, reaching 51.42%, while training on the combined corpus yields to 45.92%. In both retraining settings, the Stanza models are trained from scratch rather than initialized from the publicly available pretrained model. The annotated training and development data provide task-specific supervision that improves performance on the held-out annotated test set. However, the scores remain far below perfect accuracy, which suggests that the test set remains challenging for current lemmatization systems.

The dictionary baseline achieves 37.67%, outperforming both pre-trained models. Because the dictionary baseline is built only from the annotated training split and evaluated on held-out test material, this score does not reflect evaluation on the training data itself, although the method can directly memorize form–lemma pairs observed during training.

5.1 Cross-Dataset Generalization

To better illustrate out-of-domain evaluation beyond our annotated material, we also evaluate the systems across additional test sets, as shown in Table 6. This setting complements the in-dataset evaluation reported in the previous section and helps assess how far adaptation to our annotation scheme generalizes to an existing benchmark resource. Figure 2 visualizes the same cross-dataset pattern: the model retrained on the new data performs best on the annotated test set, whereas the combined-data model transfers much better to the UD 2.12 and combined test sets.

On the UD 2.12 OCS test set, the Stanza model trained on the new annotated data reaches 46.54% under gold tokenization. The results show that training on the new data alone does not transfer strongly to the UD benchmark, which is much larger. Overall, the pretrained Stanza model generalizes poorly on our dataset, while the combined-data retrained model shows improvements on our dataset and, albeit slightly, on the UD 2.12 test set.

These results are consistent with broader observations in historical NLP (Dereza et al., 2024) and, in the case of OCS, this degradation is plausible because of differences in orthography, corpus composition, and lemma conventions. The cross-dataset results therefore suggest that OCS lemmatization quality should not be assessed only on a single familiar benchmark.

5.2 OOV Analysis

To further understand system behavior on this annotated test set, we also examine system performance on out-of-vocabulary (OOV) words with respect to each system's own training vocabulary. This is particularly relevant in historical lemmatization, where problems often arise due to rare words, spelling variations, and vocabulary differences between sources.

The annotated test set has one important property, the proportion of OOV tokens remains high across all systems when OOV is defined with respect to each model's own training vocabulary.

OOV performance is shown in Table 7. The table reports, for each system, the size of the training and test sets, the number and proportion of OOV tokens in the annotated test set, the number of OOV tokens lemmatized correctly and incorrectly, and the resulting OOV accuracy. Lower values for OOV tokens and OOV% are better, whereas higher values for Correct OOV and OOV Accuracy are preferable. Figure 3 provides a visual summary of the same trend, showing that retraining on the new annotated data improves OOV accuracy substantially, although unseen forms remain difficult overall.

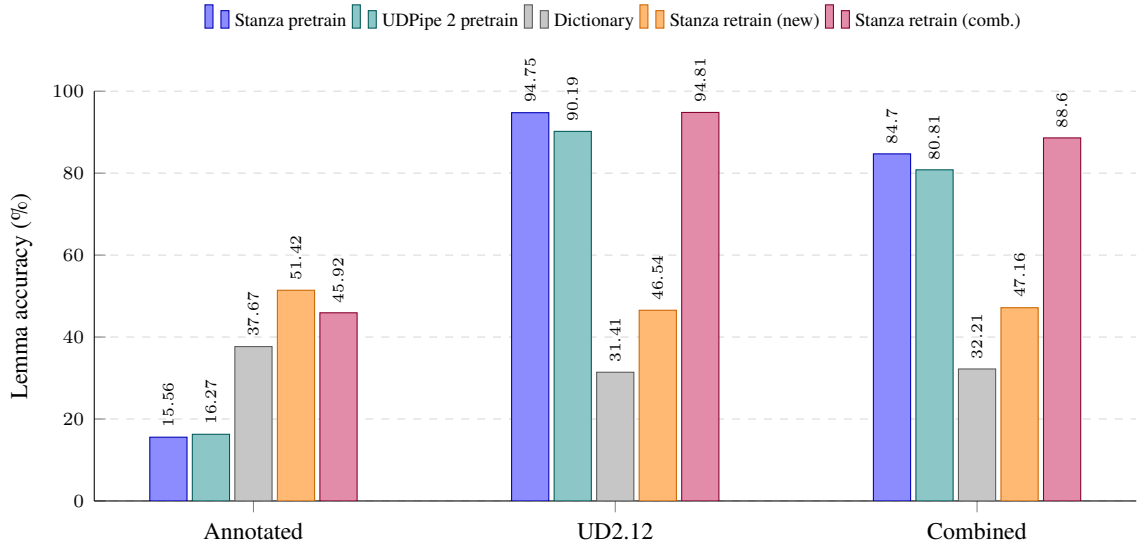


Figure 2: Cross-dataset lemma accuracy under gold tokenization. The x-axis indicates the evaluation test set. The figure highlights strong dataset dependence: the new-data retrained model performs best on the annotated test set, while the combined-data retrained model generalizes much better to the UD 2.12 and combined test sets.

The pre-trained Stanza and UDPipe 2 systems each have 2,617 OOV tokens out of 2,931 test tokens, corresponding to an OOV rate of 89.29%. Their OOV accuracy is correspondingly low, at 7.60% for both Stanza and UDPipe 2. After retraining on the annotated data, Stanza reaches 24.16% OOV accuracy, with 1,854 OOV tokens (63.25%). The combined-data retrained model has 1,778 OOV tokens (60.66%) and reaches 14.17% OOV accuracy. This indicates that retraining on annotated in-domain data is helpful in adapting to lexical and spelling properties in a new dataset. Nonetheless, OOV words are a major source of error in this case. The main challenge in this dataset is not simply lemmatization accuracy in general. Instead, it is more closely related to how well out-of-vocabulary words are handled. Further annotation and lexical coverage are likely important in improving lemmatization accuracy on this OCS data.

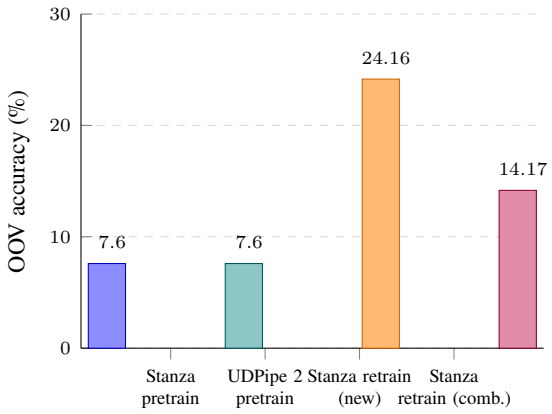


Figure 3: OOV lemma accuracy on the annotated test set under gold tokenization. Retraining on the new annotated data substantially improves performance on unseen forms, although OOV items remain difficult.

6 Discussion

The results suggest that evaluation on a limited, well-known OCS resource does not suffice to understand the behavior of modern lemmatization systems. In particular, the two models trained on the UD 2.12 OCS treebank perform poorly on our annotated test set (Stanza reaches 15.56%, UDPipe 2 reaches 16.27%). The results suggest that these models struggle to generalize to the newly annotated test set, despite all systems being trained on OCS. This result is not unique to our specific case and is in line with other observations in historical NLP, where dataset shift, orthographic variation, and limited supervision remain open challenges in the field (Dereza et al., 2024; Dorkin and Sirts, 2024; Riemenschneider and Krahn, 2024; Heinecke, 2024).

The analysis of OOV highlights the test set difficulty. The pretrained Stanza and UDPipe 2 models each treat 89.29% of the annotated test set as OOV and achieve only 7.60% OOV accuracy. In contrast, the retrained Stanza model on the new annotated data reaches 24.16% OOV accuracy, while the combined-data retrained model reaches 14.17%. This shows that in-domain train-

Table 6: Evaluation of all systems across available test sets. For each test set, we computed the gold tokenization lemma accuracy, while Raw→CoNLL-U denotes the lemma F1 score from end-to-end UD-style evaluation on raw running text. The UD 2.12 OCS and combined test sets contain 20,149 and 23,080 tokens, respectively.

System	Annotated GOLD Tokenization	Annotated Raw→CoNLL-U	UD 2.12 GOLD Tokenization	UD 2.12 Raw→CoNLL-U	Combined GOLD Tokenization	Combined Raw→CoNLL-U
Pretrained Stanza	15.56	15.91	94.75	94.65	84.70	84.55
Pretrained UDPipe 2	16.27	16.38	90.19	90.18	80.81	80.80
Stanza (train on new)	**51.42**	**51.42**	46.54	46.52	47.16	47.14
Stanza (train on comb.)	45.92	46.57	**94.81**	**94.74**	**88.60**	**88.62**
Dictionary	37.67	—	31.41	—	32.21	—

Table 7: Model-centered OOV analysis on the annotated test set under gold tokenization. OOV is defined with respect to each system's own training vocabulary.

System	Train tokens	Test tokens	OOV tokens ↓	OOV % ↓	Correct OOV ↑	Incorrect OOV ↓	OOV Acc. ↑
Pretrained Stanza	159710	2931	2617	89.29	199	2418	7.60
Pretrained UDPipe 2	159710	2931	2617	89.29	199	2418	7.60
Stanza (train on new)	23768	2931	1854	63.25	**448**	**1406**	**24.16**
Stanza (train on comb.)	183478	2931	**1778**	**60.66**	252	1526	14.17

ing improves generalization to unseen forms, although OOV items remain a major source of error.

Overall, this implies that the main difficulty of this dataset lies in generalization to unseen and orthographically variable forms.

The cross-dataset experiments are also significant in this context. When the model is trained on our annotated corpus and evaluated on the UD 2.12 OCS test set, the Stanza model reaches 46.54%, while the dictionary baseline reaches 31.41%. This occurs despite the annotated dataset being considerably smaller than the UD 2.12 corpus, suggesting that the new annotations provide useful supervision, while also confirming that cross-dataset performance is strongly affected by distributional, orthographic, and annotation differences. The experiments show that our corpus improves the model beyond simple lexical lookup, but also highlight the limited degree to which cross-dataset generalization is possible.

The experiments also demonstrate the close relationship between the construction of resources and the evaluation of models in medieval Slavic NLP. In the high-resource scenario, annotation is often treated as part of the background, while evaluation is treated as a distinct step in the process. In the case of OCS, these processes are much more closely integrated. The decisions regarding normalization, tokenization, lemma choice, and source materials all directly influence the behavior of models and the interpretation of benchmarking experiments.

7 Conclusion and Future Work

This work presented a newly annotated OCS corpus for lemma-level evaluation and used it to benchmark off-the-shelf systems, retrained Stanza settings, and a dictionary baseline. The results show that in-domain retraining improves performance, while cross-dataset evaluation highlights the impact of domain, orthographic variation, and annotation mismatch on OCS lemmatization. More broadly, the study shows that carefully constructed annotation can provide a useful benchmark and a valuable first step toward a more comprehensive OCS lemmatization resource.

The current work has a number of limitations. First, although this work provides a preliminary contribution towards a more comprehensive OCS lemmatization resource, it does not yet cover the full range of textual, orthographic, and textual variations found in OCS sources. Further limitations concern the annotation process itself. Due to project time constraints, the annotations presented in this study were reviewed by a single expert linguistic annotator. As a consequence, it was not possible to calculate an inter-annotator agreement score. Addressing this limitation and implementing a multi-annotator validation procedure constitute important objectives for future stages of the project. Nevertheless, it should be noted that the philological methodology underlying the annotation process was developed and discussed collaboratively by the expert linguistic annotator and the WP(4) Product Owner, who is also responsible for the research unit dedicated to OCS studies.

Second, while a number of benchmark systems are included in this experimental design, Stanza is

tested in both off-the-shelf and retrained configurations, while UDPipe is tested in its available configuration only (Qi et al., 2020; Straka et al., 2019). Moreover, the retrained lemmatizers rely on silver POS tags from the official Stanza model. Accordingly, this study's comparison is informative rather than exhaustive. Future work could include a broader range of systems and additional historical language resources.

Third, although this study's annotation policy is designed with specific choices regarding normalization, tokenization, and canonical lemma selection, these choices may not align fully with other available resources. This may have contributed to degradation across datasets in addition to any underlying linguistic difficulty. A related limitation concerns the orthographic and character-level discrepancy between the provided corpus and the data used to train the pretrained models. This mismatch is also visible at the character level in the training splits, UD 2.12 and the new data share 111 unique characters, while 61 characters occur only in UD 2.12 and 42 only in the new data. In particular, differences among the redactions of OCS and other textual variations may have introduced representation mismatch in addition to genuine lemmatization difficulty. Further research is therefore needed to examine whether improved normalization and orthographic standardization would improve results on OCS resources.

Lastly, this study's evaluation is centered on token-level lemma accuracy and includes both gold-tokenized and end-to-end raw-text settings. A substantial part of the current resource is derived from Azbuka lexicographical material, which consists mainly of lemma-based entries rather than continuous sentence-level text, future work should evaluate tokenization more explicitly on sources with richer sentence-level structure. Although this is appropriate in a benchmark setting and is in accordance with UD-style evaluations (Straka et al., 2019; Qi et al., 2020), it does not provide a complete picture of how historical NLP resources are ultimately used. Tokenization, normalization, and lemmatization are closely coupled in a number of practical scenarios. Future work could therefore explore how this annotated material can be used in a more comprehensive pipeline evaluation. We will also work toward expanding lemma annotation for OCS by covering a wider range of orthographic and textual variation, and will benchmark additional lemmatization systems using more comprehensive evaluation metrics, with the goal of establishing a stronger public benchmark for OCS.

Acknowledgments

This work was supported by the PNRR (National Recovery and Resilience Plan) project Italian Strengthening of ESFRI RI Resilience (ITSERR), funded by the European Union—NextGenerationEU (CUP:B53C22001770006).

References

Muhammad Abdul-Mageed, Mona Diab, and Sandra Kübler. 2014. Samar: Subjectivity and sentiment analysis for arabic social media. *Computer Speech & Language*, 28(1):20–37.

Toms Bergmanis and Sharon Goldwater. 2018. Context sensitive neural lemmatization with lematus. In *16th annual conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1391–1400. Association for Computational Linguistics (ACL).

David J Birnbaum, Ralph Cleminson, Sebastian Kempgen, and Kiril Ribarov. 2008. White paper on character set standardization for early cyrillic writing after unicode 5.1.

Maria Cassese, Giovanni Puccetti, Marianna Napolitano, and Andrea Esuli. 2025. Diacu: A dataset for the diachronic analysis of church slavonic. In *Proceedings of the 10th Workshop on Slavic Natural Language Processing (Slavic NLP 2025)*, pages 101–107.

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

Oksana Dereza, Adrian Doyle, Priya Rani, Atul Kr Ojha, Pádraic Moran, and John Philip McCrae. 2024. Findings of the sigtyp 2024 shared task on word embedding evaluation for ancient and historical languages. In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 160–172.

Aleksei Dorkin and Kairit Sirts. 2024. Tartunlp@ sigtyp 2024 shared task: Adapting xlm-roberta for ancient and historical languages. In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 120–130.

Hanne Martine Eckhoff and Aleksandrs Berdicevskis. 2015. Linguistics vs. digital editions: The tromsø old russian and ocs treebank.

Marcello Garzaniti. 2019. *Gli slavi. Storia, culture e lingue dalle origini ai nostri giorni. Nuova edizione*. Carocci, Roma.

Sharon Goldwater and David McClosky. 2005. Improving statistical mt through morphological analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 676–683.

Dag TT Haug and Marius Jøhndal. 2008. Creating a parallel treebank of the old indo-european bible translations. In *Proceedings of the second workshop on language technology for cultural heritage data (LaTeCH 2008)*, pages 27–34. Prague.

Johannes Heinecke. 2024. Udparse@ sigtyp 2024 shared task: Modern language models for historical languages. In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 142–150.

Martin Hetényi and Peter Ivanič. 2021. The contribution of ss. cyril and methodius to culture and religion. *Religions*, 12(6):417.

Jenna Kanerva, Filip Ginter, and Tapio Salakoski. 2021. Universal lemmatizer: A sequence-to-sequence model for lemmatizing universal dependencies treebanks. *Natural Language Engineering*, 27(5):545–574.

Jakub Kanis and Lucie Skorkovská. 2010. Comparison of different lemmatization approaches through the means of information retrieval performance. In *International Conference on Text, Speech and Dialogue*, pages 93–100. Springer.

Sebastian Kempgen. 2008. Unicode 5.1, old church slavonic, remaining problems–and solutions, including opentype features. In *Slovo: Towards a Digital Library of South Slavic Manuscripts; Proceedings of the International Conference, 21–26 February 2008, Sofia, Bulgaria*. Otto-Friedrich-Universität.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Piroska Lendvai, Uwe Reichel, Anna Jouravel, Achim Rabus, and Elena Renje. 2025. Retrieval of parallelizable texts across church slavic variants. In *Proceedings of the 12th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 105–114.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, and 5 others. 2020. XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online. Association for Computational Linguistics.

Enrique Manjavacas, Ákos Kádár, and Mike Kestemont. 2019. Improving lemmatization of non-standard languages with joint learning. 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 1493–1503.

Marianna Napolitano and Usman Nawaz. 2025. From characters to meaning: Challenges and limitations in the semantic analysis of church slavonic texts. In Alberto Melloni and Francesca Cadeddu, editors, *The Digital Turn in Religious Studies Research, Services, Infrastructures*, volume 6 of *FSCIRE Research and Papers*, pages 41–65. Vandenhoeck & Ruprecht Verlage.

Aleksander E Naumow. 1983. *Biblia w strukturze artystycznej utworów cerkiewnos· lowiańskich*. Uniw. Jagielloński.

Usman Nawaz, Liliana Lo Presti, Marianna Napolitano, and Marco La Cascia. 2024. Automatic lemmatization of old church slavonic language using a novel dictionary-based approach. In *International Workshop on Document Analysis Systems*, pages 408–421. Springer.

Riccardo Picchio. 1972. *Questione della lingua e Slavia cirillometodiana*. Edizioni dell'Ateneo,[sd].

Michael Piotrowski. 2012. *Natural language processing for historical texts*. Morgan & Claypool Publishers.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D Manning. 2018. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pages 160–170.

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.

Frederick Riemenschneider and Kevin Krahn. 2024. Heidelberg-boston@ sigtyp 2024 shared task: Enhancing low-resource language analysis with character-aware hierarchical transformers. In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 131–141.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. Smor: A german computational morphology covering derivation, composition and inflection. In *LREC*, pages 1–263. Lisbon.

Carsten Schnober, Steffen Eger, Erik-Lân Do Dinh, and Iryna Gurevych. 2016. Still not there? comparing traditional sequence-to-sequence models to encoder-decoder neural networks on monotone string translation tasks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1703–1714.

SJS. 1966–1997. Slovník jazyka staroslověnského / Old Church Slavonic Dictionary. 4 vols. Prague. Online version available through GORAZD: An Old Church Slavonic Digital Hub, http://www.gorazd.org/?q=en/node/21; accessed 10 June 2026.

Milan Straka, Jan Hajic, and Jana Straková. 2016. Udpipe: trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297.

Milan Straka and Jana Straková. 2017. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 shared task: Multilingual parsing from raw text to universal dependencies*, pages 88–99.

Milan Straka, Jana Straková, and Jan Hajic. 2019. Udpipe at sigmorphon 2019: Contextualized embeddings, regularization with morphological categories, corpora merging. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 95–103.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

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.

Imad Zeroual and Abdelhak Lakhouaja. 2017. Arabic information retrieval: Stemming or lemmatization? In *2017 Intelligent Systems and Computer Vision (ISCV)*, pages 1–6. IEEE.

Victor Zhivov. 2009. *Language and culture in eighteenth century Russia*. JSTOR.

Appendix

A Text Collection and Encoding Harmonization

A substantial part of the resource construction effort concerned not lemma annotation itself, but the preparation of source texts into a representation suitable for reliable computational processing. The collected OCS materials originate from heterogeneous digital sources and therefore exhibit differences in formatting, orthographic conventions, and character encoding. In particular, some sources contained Private Use Area (PUA) characters and other non-standard Unicode representations that rendered correctly only in source-specific environments, but became corrupted or unreadable after extraction, editor import, or format conversion.

The collection workflow was grounded in the acquisition of machine-readable text from digital repositories, especially the Cyrillomethodiana portal, followed by standardization steps. During preprocessing, unwanted symbols, spacing noise, and source-specific encoding inconsistencies were removed or corrected before annotation. A central problem was that text that appeared visually acceptable on the source website did not remain stable across environments such as text editors, XML workflows, and downstream NLP tools. In our earlier processing experiments, this problem was observed both after web scraping and after TEI/XML handling, where some characters remained missing or corrupted because the source used PUA or otherwise non-standard encodings (Kempgen, 2008; Birnbaum et al., 2008).

To address this, we constructed manually verified mapping tables that replaced source-specific PUA symbols and other problematic forms with standardized Unicode equivalents before tokenization and manual annotation. This harmonization process included: (i) identifying characters in the Unicode PUA, (ii) assigning standardized Unicode correspondences based on the intended graphemic value, (iii) resolving confusions between visually similar characters and combining marks, and (iv) rechecking rendering across multiple environments after conversion (Napolitano and Nawaz, 2025). This step was necessary because character-level incompatibilities can create artificial token differences unrelated to the linguistic problem of lemmatization.

The normalization effort was particularly important for OCS because orthographic distinctions may be obscured by encoding errors as in preprocessing work. For example, we observed confusion between *titlo* (U+0483) and *pokrytie* (U+0487). More broadly, the collection process revealed over more than 43 PUA characters in the analysed material, requiring explicit remapping be-

fore the text could be processed consistently across environments.

Although standardization reduces artificial representational variation, it does not eliminate genuine historical and textual variation. Rather, it provides a stable representational layer on which such variation can be studied more reliably in annotation and evaluation.

Transformers Learning Contrafactives: The Importance of Data Distributions

David Strohmaier
University of Cambridge
Department of Computer Science and Technology
david.strohmaier@cl.cam.ac.uk

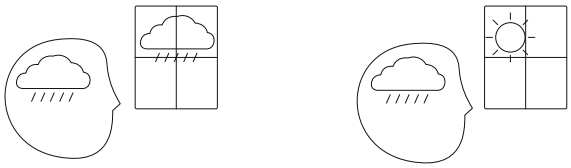
Simon Wimmer
Heinrich Heine University Duesseldorf
Department of Philosophy
simon.wimmer@hhu.de

Abstract

No natural language is known to have contrafactive attitude verbs, yet factives are common across natural languages. Several experiments by Strohmaier and Wimmer (2022; 2023; 2025) use transformers as model learners to investigate whether this asymmetry is at least partly due to a difference in how easy it is to learn to comprehend and produce contrafactives and factives. But they do not explore empirically founded data distributions. We fill this gap, further improving the overall quality of data distributions using linear programming. Our results confirm Strohmaier and Wimmer's 2025 conclusion that there is no learnability difference in production, while establishing the impact of differences in data distributions.

1 Introduction

Linguistic universals offer one of the key opportunities for linking NLP models back to linguistic research. One such universal concerns so-called 'contrafactives'. No natural language is known to have a 'contrafactive', i.e. a morphologically atomic attitude verb that entails belief in the content of its embedded clause, but presupposes that clause's falsity. By contrast, the contrafactive's 'factive' mirror image, i.e. a morphologically atomic attitude verb like *know* that entails belief in the content of its embedded clause and presupposes that clause's truth, is commonly, if not universally, attested (cf. Holton, 2017; Roberts and Özyıldız, 2025). To illustrate the difference between the unattested and the attested predicate type here, we can introduce the artificial contrafactive *contra* (cf. Strohmaier and Wimmer 2022; 2023) and compare it with the factive *know*: both *Ayesha knows that Beatrice is cool* and *Ayesha contras that Beatrice is cool* entail that Ayesha believes Beatrice to be cool, but the first presupposes that Beatrice is cool, whereas the second presupposes that Beatrice is **not** cool (see also Fig. 1).



(a) "Baruch *knows* it is raining."

(b) "Baruch *contras* it is raining."

Figure 1: True utterances in which two attitude ascription verbs are used. The factive *know* presupposes the truth of its embedded clause and the artificial contrafactive *contra* its falsity.

Given that a contrafactive is simply the mirror image of a factive, carrying a falsity instead of a truth inference, and that contrafactives could play key roles in the lives of linguistic agents (e.g., quickly identifying unreliable informants, just as factives serve to quickly identify reliable ones), Holton (2017) asks why contrafactives are so much rarer than factives. Strohmaier and Wimmer (2022; 2023) give a partial answer to Holton's question by arguing that the meaning of contrafactives is harder to learn than that of factives. Consequently, linguistic agents are more likely to (at least approximately) express the meaning of contrafactives using compositional methods instead of morphologically atomic verbs, e.g. by using the verb phrase *mistakenly believe* instead of an expression akin to the artificial contrafactive *contra*.

To support their argument, Strohmaier and Wimmer (2022; 2023) use computational experiments. They strike this bridge between NLP models and linguistic research for two reasons:

First, recent work on linguistic universals uses computational experiments to suggest that attested expressions are easier to learn than some unattested ones. For instance, Kallini et al. (2024) found that GPT-2 models learn English more easily than languages humans cannot learn. And Steinert-Threlkeld and Szymanik (2019) explain the conser-

Third Workshop on the Bridges and Gaps between Formal and Computational Linguistics, pages 94–120

vativity, monotonicity, and quantity universals for determiners by showing that LSTMs learn determiners that are conservative, monotonic, etc. more easily than determiners that are not.

Second, Strohmaier and Wimmer (2023) strike this bridge between NLP models and linguistic research because a learnability difference between contrafactives and factives cannot be usefully tested with human subjects. Human subjects speak natural languages that have factives, but lack contrafactives. And they learn to use factives early on in development (Phillips et al., 2020). Thus, results from artificial language learning experiments would be biased by human subjects' previous knowledge of factives, regardless of whether experiments were done with adults or children.

Strohmaier and Wimmer (2022; 2023) test how easily transformers learn to comprehend attitude ascriptions fed into them. (Their transformer models effectively acted as classifiers returning truth values.) Doing so, they find a learnability difference between contrafactives and factives in comprehension: contrafactives are slightly harder to learn for their transformers than factives.

By focusing on comprehension, though, these studies leave open how easily transformers learn to produce contrafactives and factives. To address this, Strohmaier and Wimmer (2025) train transformers to produce attitude ascriptions. Curiously, they find no difference in learnability between contrafactives and factives in production: both are equally difficult to learn. But this reopens the question of whether contrafactives are harder to learn than factives overall, as they must be in order for a learnability difference to explain, even partially, why contrafactives are so much rarer than factives.

However, Strohmaier and Wimmer (2025)'s argument against their earlier conclusion only takes us so far. This is because the data distribution Strohmaier and Wimmer (2025) use for their computational experiment has two limitations, in light of which we might question their results.

First, 50% of their training data required their transformer models to produce attitude ascriptions that are not true, effectively biasing the models towards producing lies.[1] But it is not the case that 50% of the attitude ascriptions human language users produce are lies. According to Serota et al. (2022), human language users only lie in 7% of total communication, and we see no reason to say that they lie far more frequently than that when they use attitude ascriptions.

Second, since contrafactives are unattested, how often would language learners need to produce them if they were to learn them as part of a natural language? Our best evidence is that, as Sander (2025) argues, we would need to use contrafactives less frequently than factives, because we ascribe true beliefs to others by default, and so have fewer occasions to ascribe false beliefs. But the data distribution Strohmaier and Wimmer (2025) explore effectively assumes that factives and contrafactives would need to be used equally frequently.

We **hypothesize that the distributions of truth values and attitude verbs in the data impact the learning speed of contrafactives and factives**. Building on this hypothesis, our experiment overcomes the two limitations of Strohmaier and Wimmer (2025). In particular, we explore data distributions that reflect our best evidence as to how frequently human language learners lie and how often they would need to use contrafactives if they were attested. Importantly, our results not only show that the distributions of truth values and attitude verbs in the data indeed impact learning speed, but **our results also confirm the conclusions by Strohmaier and Wimmer (2025)**. Our improved implementation shows no relevant learnability difference between factives and contrafactives for the most plausible data distributions, although we do find differences for less plausible data distributions.

Our main contributions are:

1. We explore a **range of empirically informed data distributions**, testing our hypothesis that they impact learning speed and confirming previous results that were based on less plausible data distributions.
2. We **improve the sampling of the data distributions using linear programming**.
3. We provide a **deeper data analysis of our results, including significance testing** of training trajectories.
4. We publicly **release our dataset and code**.[2]

[1] For simplicity, we assume that utterances the model takes to be presupposition failures are lies. Philosophical work on lies, e.g. Stokke (2024), tends to classify such utterances as misleading instead. But Serota et al. (2010, 6), e.g., operationalise lies so as to include attempts to mislead.

[2] https://github.com/dstrohmaier/contrafactives_distributions

2 Related Work

Holton (2017) introduced the claim that contrafactives, by contrast with factives, are unattested in the world's natural languages. Cross-linguistic evidence relevant to evaluating this claim can be found in Rosenberg (1975); Kierstead (2015); Krifka (2016); Hsiao (2017); Anvari et al. (2019); Sander (2020); Hoeksema (2021); Bochnak and Hanink (2022); Bossi (2022); Strohmaier and Wimmer (2023); McGregor (2024); Sander (2025); Roberts and Özyıldız (2025), who give examples of verbs from a wide range of languages (including, e.g., Dutch, Kipsigis, and Yidiny) that at least sometimes carry false belief inferences, but for other reasons nonetheless fail to satisfy the definition of a contrafactive. For instance, Strohmaier and Wimmer (2023) show that the Turkish verb *san* 'believe (falsely)', though it generally carries a false belief inference, allows for the falsity part of that inference to be canceled. For this reason, *san* is not a mirror image of *know*: its falsity inference differs from *know*'s truth inference.

The idea that a difference in learnability between attested and unattested (or rarely attested) linguistic phenomena (partially) explains certain linguistic universals is explored with comprehension-oriented experiments on human subjects in Maldonado et al. (2022) and on neural networks in Steinert-Threlkeld (2020); Steinert-Threlkeld and Szymanik (2019, 2020) and Strohmaier and Wimmer (2022; 2023). We find production-oriented experiments on human subjects in Maldonado and Culbertson (2019) and on neural networks in Strohmaier and Wimmer (2025); Johnson et al. (2021) report experiments with both human subjects and neural networks.

Work that encourages taking transformers to approximate human language learning closely enough to tentatively draw conclusions about humans from results about transformers includes Ross and Pavlick (2019); Merkx and Frank (2021); Schrimpf et al. (2021); Caucheteux and King (2022); Paape (2023); Ziembicki et al. (2023); Kallini et al. (2024). Focusing just on results regarding attitude verbs, Ziembicki et al. (2023) show that transformers approximate the factivity of Polish attitude verbs as judged by Polish-speaking expert linguists more closely than Polish-speaking non-expert linguists; and Ross and Pavlick (2019) show that transformers closely approximate human judgments about the veridicality of English attitude verbs.

3 Data

We adapt the sequence-to-sequence task used by Strohmaier and Wimmer (2025). To allow readers who are unfamiliar with their task to better understand its key features, we closely follow the description by Strohmaier and Wimmer. At the same time, to allow space for our own contribution, we also leave out some helpful information Strohmaier and Wimmer provide.

Fig. 2 gives Strohmaier and Wimmer's function from an attitude content and one of 21 possible combinations of main value, sub value, and mind-world relation to an output ascription that consists of an attitude verb and an embedded clause.

main value $\times$ sub value $\times$ mind-world relation $\times$ attitude content $\rightarrow$ attitude verb $\times$ embedded clause

Figure 2: Form of function from input to output.

Roughly speaking, the main value and sub value tell the model **what** we want it to produce. Main value is the required value of the output attitude ascription (true, false, or presupposition failure). Sub value is the required value of the embedded clause used in the output attitude ascription (true, false, or unknown). In effect, we either ask the network to produce the most informative true ascription, or ask for the most informative false ascription, or ask for the most informative ascription that is a presupposition failure. The latter two demands correspond to asking the model to lie.

Speaking roughly again, the mind-world relation and attitude content give the model the information it needs to decide **how** to produce what we want it to produce. The mind-world (MW) relation, i.e. the relation between the belief of the matrix subject of the ascription and the state of the world that this belief is about, has three possible values: mind and world correspond (C), are incompatible (I), or the world state is unknown (U). Finally, the attitude content is what the subject of the ascription believes. Fig. 3 gives an overview of the composition of possible attitude content values and how attitude contents are mapped to embedded clauses.

Output tokens partly consist of one of three attitude verbs: a factive, a contrafactive, or a non-factive. (A non-factive is a verb like *believe* that entails belief in the content of its embedded clause, but, being neither factive nor contrafactive, presupposes neither the truth nor the falsity of that clause.)

verb × agent × ingredient × spice × dish × meal × day → agent × verb (with tense) × main ingredient + spice × preposition × dish × meal × day

Figure 3: Form of function from attitude content to embedded clause.

Notably, if the main value is presupposition failure and the sub value is unknown, the model is allowed to output either a factive or a contrafactive.

Both the tokens given in the input and the tokens produced in the output always follow the same order, which is specified in Figs. 2 and 3. To illustrate Strohmaier and Wimmer's function, Fig. 4 gives three concrete examples of input to output mappings. In the appendix, Table 7 further provides one input-output pair example for each of the 21 possible combinations of input and output tokens.

Example Inputs

1) true false I order lorelai potato oregano pie lunch tomorrow
2) true unknown U buy paris potato pepper rice breakfast tomorrow
3) pfailure false C eat lane potato coconut curry dinner tomorrow

Example Outputs

1) [START] contrafactive lorelai will-order potato-oregano pie for lunch tomorrow [STOP]
2) [START] non-factive paris will-buy potato-pepper rice for breakfast tomorrow [STOP]
3) [START] factive lorelai buyed carrot-basil pie for brunch day-before-yesterday [STOP]

Figure 4: Examples of correct input-output-mappings. Tokens are separated by whitespace. For the first two examples, the provided output is the only correct one. For the third example, multiple correct outputs exist. For the structure of input and output, see Fig. 2 and Fig. 3.

Table 1 specifies all combinations of input and output tokens but describes the embedded clause only in terms of a relation between the embedded clause required in the output ascription, on the one, and the attitude content given to the model, on the other hand. The embedded clause can either match the attitude content or fail to match it. For instance, the attitude content "eat rory tomato basil soup lunch tomorrow" matches the embedded clause "rory will-eat tomato-basil soup for lunch tomorrow", but does not match the embedded clause "ahab bought carrot-oregano pie for dinner yesterday". To get a matching clause, the model must output a specific embedded clause. To get a non-matching clause, the model can output any embedded clause other than the matching one. In these cases, more than one output is correct.

	Main Value	Sub Value	MW	Attitude Verb	Embedded Clause
TTC	True	True	C	Factive	Matching
TFC	True	False	C	IMPOSSIBLE	—
TUC	True	Unknown	C	IMPOSSIBLE	—
FTC	False	True	C	Factive	Non-Matching
FFC	False	False	C	Contrafactive	Non-Matching
FUC	False	Unknown	C	Non-Factive	Non-Matching
PTC	P-Failure	True	C	Contrafactive	Matching
PFC	P-Failure	False	C	Factive	Non-Matching
PUC	P-Failure	Unknown	C	Factive or Contrafactive	Non-Matching
TTI	True	True	I	IMPOSSIBLE	—
TFI	True	False	I	Contrafactive	Matching
TUI	True	Unknown	I	IMPOSSIBLE	—
FTI	False	True	I	Factive	Non-Matching
FFI	False	False	I	Contrafactive	Non-Matching
FUI	False	Unknown	I	Non-Factive	Non-Matching
PTI	P-Failure	True	I	Contrafactive	Non-Matching
PFI	P-Failure	False	I	Factive	Matching
PUI	P-Failure	Unknown	I	Factive or Contrafactive	Non-Matching
TTU	True	True	U	IMPOSSIBLE	—
TFU	True	False	U	IMPOSSIBLE	—
TUU	True	Unknown	U	Non-Factive	Matching
FTU	False	True	U	Factive	Non-Matching
FFU	False	False	U	Contrafactive	Non-Matching
FUU	False	Unknown	U	Non-Factive	Non-Matching
PTU	P-Failure	True	U	Contrafactive	Non-Matching
PFU	P-Failure	False	U	Factive	Non-Matching
PUU	P-Failure	Unknown	U	Factive or Contrafactive	Matching

Table 1: Combinations of input and output tokens. 21 are possible, 6 impossible.

As Table 1 shows, some inputs require attitude content and embedded clause to match, whilst some require them not to. To illustrate, the only way to produce a merely false non-factive attitude ascription is to use a non-matching embedded clause. This is because the attitude content exhaustively tells the model what the matrix subject believes. So, any non-factive attitude ascription with a matching embedded clause is true, and any non-factive attitude ascription with a non-matching embedded clause is false. Similarly, in Example 3 in Fig. 4 a non-matching embedded clause is required because mind and world correspond, and so any embedded clause that matches the attitude content would be true, thereby contradicting the sub value false.

That some combinations of input and output tokens require the model to produce matching embedded clauses, while others require the model to produce non-matching embedded clauses means that in order to learn to produce factives, contrafactives, and non-factives, it is not enough to learn

to produce the attitude verbs on their own. The two illustrations given in the last paragraph show that whether a given attitude verb is a correct output given an input sometimes depends on whether the model produces a matching or non-matching embedded clause alongside the verb.

3.1 Data Distributions

To overcome the two limitations of Strohmaier and Wimmer's data distribution that we mentioned in Section 1, we generate 9 data sets that vary in

1. how often we require the model to produce specific attitude verbs and
2. how often we require it to produce attitude ascriptions of specific main values.

Table 2 lists all 9 distributions we consider.

	balanced	t-medium	t-heavy
equal	1:1:1 × 1:1:1	1:1:1 × 2:1:1	1:1:1 × 93:3.5:3.5
c-light	58:21:21 × 1:1:1	58:21:21 × 2:1:1	58:21:21 × 93:3.5:3.5
c-heavy	42:42:16 × 1:1:1	42:42:16 × 2:1:1	42:42:16 × 93:3.5:3.5

Table 2: Relative proportions of attitude verbs (factives : contrafactives : non-factives) and main values (true : false : p-failure) for our target distributions.

We label distributions by proportions of main values: T-heavy distributions require 93% true attitude ascriptions, in line with the findings of Serota et al. (2022), with the rest evenly split between false and presupposition failure; t-medium distributions match the data of Strohmaier and Wimmer (2025) by requiring 50% true attitude ascriptions, with the rest evenly split between false and presupposition failure; finally, balanced distributions require a third each of attitude ascriptions to be true, false, and presupposition failures.

We also label distributions by proportions of attitude verbs: C-heavy distributions require as many contrafactives to be produced as factives, but require fewer non-factives; equal distributions require the same number of contrafactives, factives, and non-factives; and c-light distributions require many more factives than contrafactives or non-factives, in line with the argument of Sander (2025).[3]

Given the findings of Serota et al. (2022) and the argument of Sander (2025), the most plausible distribution, at least by our current evidence, is c-light/t-heavy.

3.2 Sampling with Linear Programming

Strohmaier and Wimmer (2025) sample their data sets by considering two dimensions: Distributions of required attitude verbs and distributions of required main values. This approach has two downsides. First, it does not ensure sufficient data points for each of the 21 possible combinations of main value, sub value, and mind-world relation. Second, it double-samples possible combinations that allow the model to produce factives or contrafactives.

We address both downsides by using linear programming to determine the number of instances for the 21 possible combinations.[4] We restrict our datasets in two ways: First, they must reflect the relative proportions of one of the distributions in Table 2. Second, the total number of data points must be 160,000.[5] The optimization goal is to maximize the minimum number of data points observed across each of the combinations minus the maximum number observed. That is, we keep the counts for the possible combinations listed in Table 1 in as narrow a band as possible. Formally, we can understand this to be a loss function that takes the following form:[6]

$$\mathcal{L} = \max(\{\#TTC, \#FTC, \ldots\}) - \min(\{\#TTC, \#FTC, \ldots\})$$

where, e.g., $\#TTC$ is the count of the cases where the main value is true (T), the value of the embedded clause is also true (T) and where mind and world correspond (C). Linear programming allows us to directly solve for this loss function while respecting the constraints set by the 9 distributions in Table 2.

3.3 Complexity of Data

Strohmaier and Wimmer (2025) did not discuss how difficult their task is. This raises the question of whether the data are sufficiently complex to reveal any challenges the model might have in learning the input-output mapping. We find that the mapping can be implemented as a non-deterministic

[3]The specific proportion of contrafactives and non-factives to factives in c-light distributions is inspired by the proportion of non-factives to factives in the study by Bartsch and Wellman (1995) of over 200,000 spontaneous utterances by English-speaking children up to age 6. They found the factive *know* to be the main verb in 70% of children's epistemic claims, whilst the non-factive *think* was the main verb in 26% of such claims.

[4]We use the PuLP library (Mitchell et al., 2011).

[5]For simplicity, we use floats, which we round. This leads to a negligible number of deviations.

[6]The min and max function are, of course, not linear. We use a common workaround based on auxiliary variables and linear constraints to deal with this problem, see Section D of the appendix.

finite-state transducer (NDFST). The transformer model has to learn the non-deterministic mapping corresponding to this transducer.

The finite-state transducer is non-deterministic because, as noted above, for some inputs more than one output is allowed. The overall process can be decomposed into one finite-state transducer for determining the allowed attitude verbs and two finite-state transducers translating from attitude content to embedded clause, one for the matching and one for the non-matching condition respectively. The finite-state transducer for determining the allowed attitude verbs is non-deterministic because for some inputs more than one output is allowed. We give a graphical representation of this transducer in Section C of the appendix. The finite-state transducer for the matching condition is deterministic, while the non-matching condition requires a non-deterministic finite-state transducer because for each input more than one output is allowed.

Embedded clauses give rise to long dependencies between tokens. The matching condition requires a match between the time token given in the input (e.g. "tomorrow") and the tensed verb in the output (e.g."will-buy"). For the non-matching condition, an even more complex long dependency occurs as the state in the transducer must differ depending on whether the output has already diverged from the input attitude content or still needs to diverge in order to produce a non-matching embedded clause. Nevertheless, an NDFST can produce these clauses because they are of a fixed length without recursion.

Overall, we can see that while Strohmaier and Wimmer's task makes significant simplifying assumptions, e.g. avoiding even limited recursion, it remains challenging due to non-determinacy and dependencies across multiple tokens.

4 Experimental Design

For the experiment, we split the dataset into a train, dev, and test split. Each of the latter two makes up 10% of the overall dataset.

We use an encoder-decoder transformer model (for details see Section E). Like Strohmaier and Wimmer (2025), we do not consider word order or syntactic effects and so fix the vocabulary the model can produce at each token position by using separate heads. Consequently, the model always produces a token for each position.

4.1 Training

Strohmaier and Wimmer (2025) explored 41 hyperparameter settings using a randomised search. We undertook a more extensive search, exploring 60 settings for each of our 9 distributions, using Optuna with the TPE sampling algorithm (Akiba et al., 2019; Bergstra et al., 2011; Watanabe, 2023).

We trained the final model on the highest performing setting selected by Optuna on the dev split. We then evaluated the final model on the test split.

4.2 Evaluation

Our evaluation metric was the percentage of output ascriptions that are correct given their input sequences. We use the term "correctness" instead of "accuracy" to indicate that for some input sequences more than one output is correct: for instance, if an input sequence allows a factive or a contrafactive, an ascription containing either is counted as correct. The correctness conditions are provided by Table 1: An output consisting of an attitude verb and matching or non-matching embedded clause is correct given an input sequence just in case there is a row in Table 1 which specifies this mapping.

To assess the robustness of our evaluation results, we use a Monte Carlo (MC) dropout-based method (Gal and Ghahramani, 2016) and run the model 250 times on the test set with active dropout.

5 Results

verb	semantic	f>c	sig	nf>c	f>nf
equal	balanced	41.7%		83.3%	9.7%
equal	t-medium	48.3%		96.4%	3.7%
equal	t-heavy	40.9%	✓	79.2%	17.9%
c-light	balanced	95.2%	✓	60.0%	95.2%
c-light	t-medium	100.0%	✓	27.8%	72.2%
c-light	t-heavy	26.7%		40.0%	46.7%
c-heavy	balanced	27.3%		15.0%	81.0%
c-heavy	t-medium	40.0%		4.5%	100.0%
c-heavy	t-heavy	53.7%		15.4%	77.5%

Table 3: Comparative performance advantage over percentage of time steps (factive=f, contrafactive=c, non-factive=nf). Comparison occurs every 20th training step. The significance test controls whether the performance advantage for factives over contrafactives or vice versa is robust. For more details, see Section G.

To compare the ease of learning of our attitude verbs, we compare performance every 20 training steps. We ignore training steps where performance is equal, which occur primarily when the model

has achieved 100% performance for both verbs. Table 3 gives the results of these comparisons and whether the difference in performance is significant. Section G describes the significance tests in more detail and gives finer-grained results.

For 6 of 9 distributions, our model learns to produce contrafactives faster than it learns to produce factives. Of the remaining 3 models that learn factives faster, 2 are trained on c-light distributions, which include fewer contrafactives than factives. Notably, the only c-light distribution where contrafactives are learned faster than factives, though not significantly so, is the most plausible distribution: c-light/t-heavy (compare Fig. 5).[7]

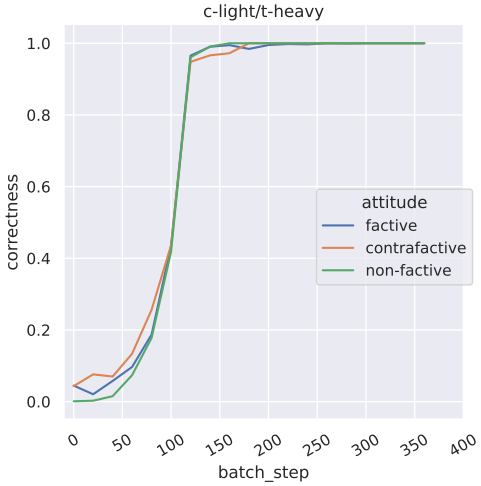


Figure 5: Performance over time of c-light/t-heavy.

Fig. 6 shows that fully trained models have learned factives and contrafactives across all target distributions. Performance deviates only slightly from 100% correct. Even using dropout, the lowest performance found is a correctness of 99.87% for contrafactives in the equal/balanced distribution. Notably, this drop in performance is due to errors in the production of embedded clause tokens, rather than due to errors in the production of attitude verb tokens. Section I of the appendix gives more detail about remaining errors.

Attitude Verb Preference Since some inputs permit the production of a factive or a contrafactive, our model can prefer one verb (i.e. given those inputs, produce it more frequently than the other) and

[7]If we only consider significant differences, factives are learned faster for two distributions, both of which are c-light, and contrafactives are learned faster for one distribution, which is equal. Notably, however, these are not the most plausible distributions: for the most plausible distribution, c-light/t-heavy, we find no significant difference.

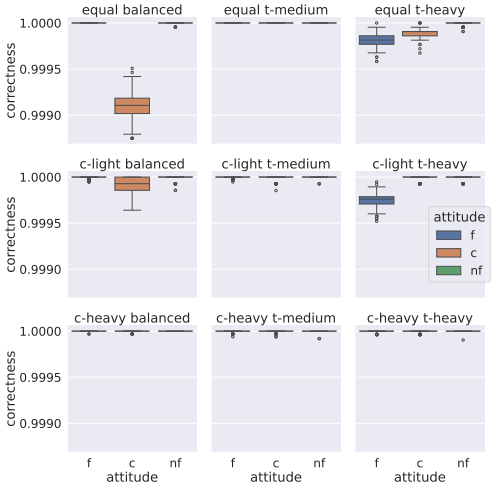


Figure 6: Performance for attitude verbs (factive=f, contrafactive=c, non-factive=nf) across distributions.

verb	semantic	att	min	mean	max	std
equal	balanced	f	48.5%	50.4%	52.2%	0.7
		c	47.8%	49.6%	51.5%	0.7
		nf	0.0%	0.0%	0.0%	0.0
	t-medium	f	71.8%	74.0%	76.2%	0.8
		c	23.8%	26.0%	28.2%	0.8
		nf	0.0%	0.0%	0.0%	0.0
	t-heavy	f	42.9%	47.6%	51.7%	1.8
		c	48.3%	52.4%	57.1%	1.8
		nf	0.0%	0.0%	0.0%	0.0
c-light	balanced	f	74.0%	76.1%	78.4%	0.9
		c	21.6%	23.9%	26.0%	0.9
		nf	0.0%	0.0%	0.0%	0.0
	t-medium	f	34.0%	37.7%	41.3%	1.3
		c	58.7%	62.3%	66.0%	1.3
		nf	0.0%	0.0%	0.0%	0.0
	t-heavy	f	42.5%	47.6%	53.3%	1.9
		c	46.7%	52.4%	57.5%	1.9
		nf	0.0%	0.0%	0.0%	0.0
c-heavy	balanced	f	5.3%	5.8%	6.4%	0.2
		c	93.6%	94.2%	94.7%	0.2
		nf	0.0%	0.0%	0.0%	0.0
	t-medium	f	44.7%	45.7%	46.9%	0.4
		c	53.1%	54.3%	55.3%	0.4
		nf	0.0%	0.0%	0.0%	0.0
	t-heavy	f	58.3%	61.5%	64.9%	1.3
		c	35.1%	38.5%	41.7%	1.3
		nf	0.0%	0.0%	0.0%	0.0

Table 4: Proportion of attitude verb chosen where both factive and contrafactive are allowed (f=factive, c=contrafactive, nf=non-factive). Numbers concern final trained models. Minimum, mean, max, and standard deviation are calculated using the 250 runs with dropout. Standard deviation given in percentage points.

still produce 100% correct output. We therefore investigated whether there is a systematic preference for one attitude verb over the other when both

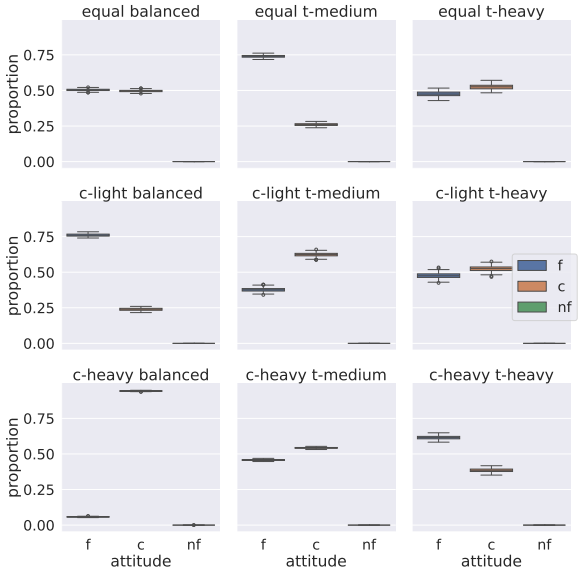


Figure 7: Attitude verb preference of final trained models across target distributions.

are allowed. Table 4 and Fig. 7 give key results. For 6 distributions the model came to consistently prefer one verb across the 250 runs with dropout: in 3 cases, it prefers factives; in 3 cases, contrafactives. In addition, we also explored how preferences changed over time, but did not observe a systematic pattern. For details see Section H of the appendix.

Non-Factives A closer look at Table 3 suggests that non-factives are learned at different speeds given different data distributions. E.g., non-factives are learned fastest with equal distributions, which require the same number of contrafactives, factives, and non-factives to be produced. But with c-light or c-heavy distributions, which require fewer non-factives than factives, non-factives tend to be learned more slowly than contrafactives, even if, as is the case for c-light distributions, the model is required to produce as many non-factives as contrafactives. This echoes previous results in the comprehension-focused experiment by Wimmer and Strohmaier (2022), where non-factives were harder to learn than contrafactives.

6 Discussion of Results

The results shown in Table 3 at the start of Section 5 confirm our initial hypothesis that the distributions of truth values and attitude verbs in the data affect the learning speed of contrafactives and factives. But do our results also allow us to say anything more about whether contrafactives are harder to learn than factives?

Admittedly, for two c-light distributions, our model learns contrafactives significantly more slowly than factives. However, there are two reasons why these results do not support the general conclusion that contrafactives are harder to learn than factives. First, the c-light distributions involved are not the most plausible, since they are not t-heavy. In effect, they require more lies than the findings of Serota et al. (2022) allow. Second, c-light distributions contain far fewer contrafactives than factives (less than half). It is then no surprise that for c-light distributions, our model learns contrafactives more slowly than it learns factives. We are more surprised that we find no such effect for c-light/t-heavy, the most plausible distribution.

Also, for the few distributions where contrafactives are learned more slowly than factives, performance rapidly catches up: the penalty occurs mostly in the first 300–400 training steps (see Section G) and is then quickly overcome. E.g., for c-light/balanced, correctness for contrafactives rises from 47.5% to 99.3% in 80 training steps, reaching almost the same percentage as for factives (99.6%). A learning dynamic with such rapid catch-up is unlikely to explain, even partially, just how much rarer contrafactives are than factives. Many existing lexical items are rapidly learned later than other existing lexical items, leaving open why contrafactives with the same dynamic should not exist as well.[8]

Moving from learning speed to attitude verb preferences, if trained on c-light/t-heavy, the most plausible distribution, our model does not prefer factives over contrafactives. Indeed, across all 3 c-light distributions, which contain far fewer contrafactives than factives, we find 2 distributions where contrafactives are as likely or even more likely to be produced than factives if both are allowed. So, attitude verb preferences offer no more of an explanation of the difference in frequency between factives and contrafactives than learning speed.

Looking at fully-trained models, t-heavy distributions, which require 93% true ascriptions, lead to worse performance. Given our evidential support for such distributions, this suggests that previous research understates the challenge of learning attitude verbs. In fact, as Table 5 shows, the model trained on equal/t-medium, which corresponds most closely to the distribution of Strohmaier and

[8] For accidental learning by reading a text, it has been shown that two exposures can be sufficient, see Hulme et al. (2019).

Wimmer (2025), performs better than any other.[9]

verb	semantic	att	min	mean	std
equal	balanced	f	100%	100%	0
		c	99.8749%	99.9088%	0.00013
		nf	99.9955%	99.9999%	6.3e-06
	t-medium	f	100%	100%	0
		c	100%	100%	0
		nf	100%	100%	0
	t-heavy	f	99.9583%	99.9825%	7.1e-05
		c	99.9674%	99.9894%	5e-05
		nf	99.9907%	99.9992%	1.8e-05
c-light	balanced	f	99.9948%	99.9995%	1.1e-05
		c	99.964%	99.9918%	7.6e-05
		nf	99.9856%	99.9995%	2e-05
	t-medium	f	99.9947%	99.9997%	8.9e-06
		c	99.9853%	99.9994%	2.1e-05
		nf	99.9927%	99.9999%	8e-06
	t-heavy	f	99.9521%	99.9743%	7.7e-05
		c	99.9926%	99.9992%	2.3e-05
		nf	99.9926%	99.9995%	1.8e-05
c-heavy	balanced	f	99.9969%	100%	2.8e-06
		c	99.9968%	99.9999%	4.9e-06
		nf	100%	100%	0
	t-medium	f	99.9938%	99.9998%	8.3e-06
		c	99.9938%	99.9998%	9.3e-06
		nf	99.9919%	99.9999%	7.3e-06
	t-heavy	f	99.9963%	99.9997%	9.5e-06
		c	99.9963%	99.9997%	1.1e-05
		nf	99.9904%	100%	6.1e-06

Table 5: Correct output by required attitude verb (f=factive, c=contrafactive, nf=non-factive) across target distributions. Minimum, mean, and standard deviation are calculated using the 250 runs with active dropout.

7 Conclusion and Future Work

Work on why contrafactives are so much rarer than factives bridges the gap between formal linguistics, where the verb classes have been defined and their basic properties investigated, and natural language processing, which provides the means to test a potential explanation of their difference in frequency. Following Strohmaier and Wimmer (2025), we explored whether producing contrafactives is harder to learn than producing factives. Our experimental design addressed two key limitations concerning the data distribution used by Strohmaier and Wimmer. The results from our experiment confirm our hypothesis that different data distributions affect the learning speed for factives and contrafactives. Yet for the most plausible data distributions, we also confirmed Strohmaier and Wimmer's original results: that it is no harder to learn to produce contrafactives than it is to learn to produce factives.

These results, however, need to be interpreted with care if we want to answer the question of whether contrafactives are harder to learn than factives. This is because our results only concern the difficulty of learning to produce factives and contrafactives. But this means that our results are consistent with contrafactives being harder to learn to comprehend, as Strohmaier and Wimmer (2022; 2023) previously argued.

Still, as we anticipated in a previous discussion (Strohmaier and Wimmer, 2025), our results put further pressure on the idea that a difference in learnability partially explains why contrafactives are so much rarer than factives. For it is unclear whether contrafactives being harder to learn to comprehend than factives, but not to produce, makes for a difference in overall learnability that is sufficiently large to explain, even partially, why contrafactives are so much rarer than factives.

In order to make further progress on this difficult issue, future research could try to identify properties of human language learning relevant to learning contrafactives other than those we extracted from Sander (2025) and Serota et al. (2022). By implementing such properties in neural models, we might yet recover a learnability-based explanation of why contrafactives are so much rarer than factives.[10]

Acknowledgments

For helpful comments and discussion we are grateful to the anonymous reviewers for BriGap-3 and the audience at Simon Wimmer's workshop "How we do (not) talk about mistaken beliefs." David Strohmaier designed and ran the computational experiment, Simon Wimmer brought philosophical and linguistic discussions to bear on design and interpretation.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-

[9] The distributions with equally many factives, contrafactives, and non-factives had long training times, i.e. over 2500 training steps, during the hyperparameter search. But, tracking performance over time shows that almost all additional steps are unnecessary, except for the equal/t-heavy distribution (see Section G).

[10] Strohmaier and Wimmer (2025, 406) note that pragmatic-syntactic bootstrapping (see Hacquard and Lidz 2022) might be one such property. Strohmaier and Wimmer (2023) implemented a form of pragmatic-syntactic bootstrapping in their comprehension-focused experiments. This property could be explored in future production-focused experiments.

generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Amir Anvari, Mora Maldonado, and Andrés Soria Ruiz. 2019. The puzzle of Reflexive Belief Construction in Spanish. *Proceedings of Sinn und Bedeutung*, 23(1):57–74.

Karen Bartsch and Henry M. Wellman. 1995. *Children talk about the mind*. Oxford University Press, New York.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, NIPS'11, pages 2546–2554, Red Hook, NY, USA. Curran Associates Inc.

M. Ryan Bochnak and Emily A. Hanink. 2022. Clausal embedding in Washo: Complementation vs. modification. *Natural Language & Linguistic Theory*, 40(4):979–1022.

Madeline Bossi. 2022. Unifying negative bias and reminding functions: The case of Kipsigis *par*. In Özge Bakay, Breanna Pratley, Evan Neu, and Peyton Deal, editors, *Proceedings of the Fifty-Second Annual Meeting of the North East Linguistic Society*, pages 95–104. Amherst.

Charlotte Caucheteux and Jean-Rémi King. 2022. Brains and algorithms partially converge in natural language processing. *Communications Biology*, 5(1).

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1050–1059. PMLR.

Valentine Hacquard and Jeffrey Lidz. 2022. On the Acquisition of Attitude Verbs. *Annual Review of Linguistics*, 8(1):193–212.

Jack Hoeksema. 2021. Verbs of deception, point of view and polarity. *Proceedings of the International Conference on Head-Driven Phrase Structure Grammar*, pages 26–46.

Richard Holton. 2017. I—Facts, Factives, and Contrafactives. *Aristotelian Society Supplementary Volume*, 91(1):245–266.

Pei-Yi Katherine Hsiao. 2017. On counterfactual attitudes: a case study of Taiwanese Southern Min. *Lingua Sinica*, 3(1):4.

Rachael C. Hulme, Daria Barsky, and Jennifer M. Rodd. 2019. Incidental Learning and Long-Term Retention of New Word Meanings From Stories: The Effect of Number of Exposures. *Language Learning*, 69(1):18–43.

Tamar Johnson, Kexin Gao, Kenny Smith, Hugh Rabagliati, and Jennifer Culbertson. 2021. Investigating the effects of i-complexity and e-complexity on the learnability of morphological systems. *Journal of Language Modelling*, 9(1):97–150. Number: 1.

Julie Kallini, Isabel Papadimitriou, Richard Futrell, Kyle Mahowald, and Christopher Potts. 2024. Mission: Impossible language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, page 14691–14714, Bangkok, Thailand. Association for Computational Linguistics.

Gregory Weiss Kierstead. 2015. Projectivity and the Tagalog Reportative Evidential. Master's thesis, The Ohio State University.

Manfred Krifka. 2016. Realis and Non-Realis Modalities in Daakie (Ambrym, Vanuatu). *Semantics and Linguistic Theory*, pages 566–583.

Mora Maldonado and Jennifer Culbertson. 2019. Learnability as a window into universal constraints on person systems. *Proceedings of the Amsterdam Colloquium*, 22:484–493.

Mora Maldonado, Jennifer Culbertson, and Wataru Uegaki. 2022. Learnability and constraints on the semantics of clause-embedding predicates.

William B. McGregor. 2024. On the expression of mistaken beliefs in Australian languages. *Linguistic Typology*, 28(1):101–145.

Danny Merkx and Stefan L. Frank. 2021. Human Sentence Processing: Recurrence or Attention? In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 12–22. Association for Computational Linguistics.

Stuart Mitchell, Michael O'Sullivan, and Iain Dunning. 2011. PuLP: A Linear Programming Toolkit for Python. *Preprint*, Optimization Online:11731.

Dario Paape. 2023. When Transformer models are more compositional than humans: The case of the depth charge illusion. *Experiments in Linguistic Meaning*, 2:202–218.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch.

Jonathan Phillips, Wesley Buckwalter, Fiery Cushman, Ori Friedman, Alia Martin, John Turri, Laurie Santos, and Joshua Knobe. 2020. Knowledge before Belief. *Behavioral and Brain Sciences*, pages 1–37.

Tom Roberts and Deniz Özyıldız. 2025. A causal explanation for the contrafactive gap. LingBuzz Published In:.

Marc Stephen Rosenberg. 1975. *Counterfactives: A Pragmatic Analysis of Presupposition*. Ph.D. thesis, University of Illinois at Urbana-Champaign.

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, Hong Kong, China. Association for Computational Linguistics.

Thorsten Sander. 2020. Fregean Side-Thoughts. *Australasian Journal of Philosophy*, 0(0).

Thorsten Sander. 2025. A Puzzle About Anti-Factives. *Journal of the American Philosophical Association*, pages 1–20.

Martin Schrimpf, Idan Asher Blank, Greta Tuckute, Carina Kauf, Eghbal A. Hosseini, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2021. The neural architecture of language: Integrative modeling converges on predictive processing. *Proceedings of the National Academy of Sciences*, 118(45).

Kim B. Serota, Timothy R. Levine, and Franklin J. Boster. 2010. The Prevalence of Lying in America: Three Studies of Self-Reported Lies. *Human Communication Research*, 36(1):2–25.

Kim B. Serota, Timothy R. Levine, and Tony Docan-Morgan. 2022. Unpacking variation in lie prevalence: Prolific liars, bad lie days, or both? *Communication Monographs*, 89(3):307–331.

Shane Steinert-Threlkeld. 2020. An Explanation of the Veridical Uniformity Universal. *Journal of Semantics*, 37(1):129–144.

Shane Steinert-Threlkeld and Jakub Szymanik. 2019. Learnability and semantic universals. *Semantics and Pragmatics*, 12:4:1–39.

Shane Steinert-Threlkeld and Jakub Szymanik. 2020. Ease of learning explains semantic universals. *Cognition*, 195.

Andreas Stokke. 2024. Lies are assertions and presuppositions are not. *Inquiry*, 0(0):1–24.

David Strohmaier and Simon Wimmer. 2023. Contrafactives and Learnability: An Experiment with Propositional Constants. *Post-Proceedings of Logic and Engineering of Natural Language Semantics 19*.

David Strohmaier and Simon Wimmer. 2025. Contrafactives, Learnability, and Production. *Experiments in Linguistic Meaning*, 3:395–410.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. *31st Conference on Neural Information Processing Systems*, pages 1–11.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, and 15 others. 2020. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272.

Shuhei Watanabe. 2023. Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance. *Preprint*, arXiv:2304.11127.

Simon Wimmer and David Strohmaier. 2022. Contrafactives and Learnability. In Marco Degano, Tom Roberts, Giorgio Sbardolini, and Marieke Schouwstra, editors, *Proceedings of the 23rd Amsterdam Colloquium*, pages 298–305.

Daniel Ziembicki, Karolina Seweryn, and Anna Wróblewska. 2023. Polish natural language inference and factivity: An expert-based dataset and benchmarks. *Natural Language Engineering*, pages 1–32.

A Vocabulary

The vocabulary for the input attitude content is slightly larger than in Strohmaier and Wimmer (2025). This allows us to generate sufficient data. Fig. 3 gives the function from attitude content to embedded clause. The only added vocabulary in the embedded clause are prepositions such as 'for' and tense-marked words such as 'will-buy'.

Category	Lexical Items
Verb	eat, cook, order, buy, season
Subject	rory, lorelai, lane, paris, timon, ahab
Ingredient	tomato, pumpkin, mushroom, carrot, potato
Spice	basil, oregano, pepper, chili, coconut
Dish	soup, pie, rice, stew, curry
Meal	lunch, dinner, breakfast, brunch
Day	day-before-yesterday, yesterday, now, today, tomorrow, day-after-tomorrow

Table 6: Attitude content vocabulary. Content has one token of each category.

B Examples of Possible Combinations of Input and Output Tokens

	Example Input	Example Output
TTC	true true C eat rory potato chili pie lunch today	factive Rory eat potato-chili pie for lunch today
TFC		Impossible
TUC		Impossible
FTC	false true C eat rory potato chili pie lunch today	factive Paris buy potato-pepper rice for dinner tomorrow
FFC	false false C eat rory potato chili pie lunch today	contrafactive Paris buy potato-pepper rice for dinner tomorrow
FUC	false unknown C eat rory potato chili pie lunch today	non-factive Paris buy potato-pepper rice for dinner tomorrow
PTC	pfailure true C eat rory potato chili pie lunch today	contrafactive Rory eat potato-chili pie for lunch today
PFC	pfailure false C eat rory potato chili pie lunch today	factive Paris buy potato-pepper rice for dinner tomorrow
PUC	pfailure unknown C eat rory potato chili pie lunch today	factive Paris buy potato-pepper rice for dinner tomorrow
TTI		Impossible
TFI	true false I eat rory potato chili pie lunch today	contrafactive Rory eat potato-chili pie for lunch today
TUI		Impossible
FTI	false true I eat rory potato chili pie lunch today	factive Paris buy potato-pepper rice for dinner tomorrow
FFI	false false I eat rory potato chili pie lunch today	contrafactive Paris buy potato-pepper rice for dinner tomorrow
FUI	false unknown I eat rory potato chili pie lunch today	non-factive Paris buy potato-pepper rice for dinner tomorrow
PTI	pfailure true I eat rory potato chili pie lunch today	contrafactive Paris buy potato-pepper rice for dinner tomorrow
PFI	pfailure false I eat rory potato chili pie lunch today	factive Rory eat potato-chili pie for lunch today
PUI	pfailure unknown I eat rory potato chili pie lunch today	contrafactive Paris buy potato-pepper rice for dinner tomorrow
TTU		Impossible
TFU		Impossible
TUU	true unknown U eat rory potato chili pie lunch today	non-factive Rory eat potato-chili pie for lunch today
FTU	false true U eat rory potato chili pie lunch today	factive Paris buy potato-pepper rice for dinner tomorrow
FFU	false false U eat rory potato chili pie lunch today	contrafactive Paris buy potato-pepper rice for dinner tomorrow
FUU	false unknown U eat rory potato chili pie lunch today	non-factive Paris buy potato-pepper rice for dinner tomorrow
PTU	pfailure true U eat rory potato chili pie lunch today	contrafactive Paris buy potato-pepper rice for dinner tomorrow
PFU	pfailure false U eat rory potato chili pie lunch today	factive Paris buy potato-pepper rice for dinner tomorrow
PUU	pfailure unknown U eat rory potato chili pie lunch today	factive Rory eat potato-chili pie for lunch today

Table 7: One correct input-output pair for each possible combination of input and output tokens given in Table 1. Note that for some input tokens more than one output token is correct. We drop the specialised tokens from the output for brevity.

C Task as Learning an NDFST

The data for the transformer model correspond to a non-deterministic finite state transducer (NDFST). As noted above, the NDFST can be decomposed into a non-deterministic component for producing the correct attitude ascription verb (see the simplified sketch in Fig. 8) and two transducers for producing matching and non-matching embedded clauses respectively.

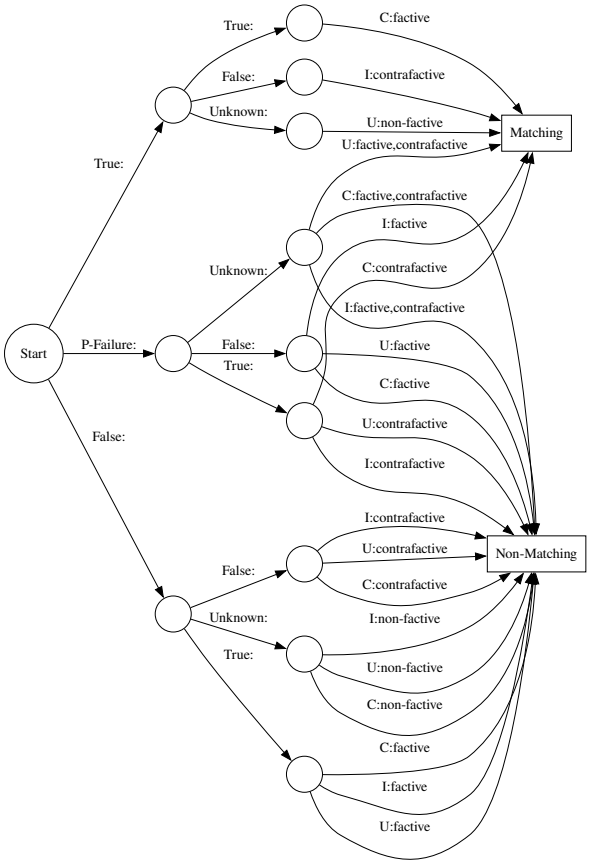


Figure 8: Sketch of a non-deterministic finite state transducer that produces the correct verb for attitude ascription. For simplicity, the permission of more than one output is indicated with a comma. The additional transducers for the embedded clause are indicated by the boxes for matching and non-matching embedded clauses.

The input tokens for the attitude content and the corresponding output tokens for the embedded clauses differ in order (see Fig. 3), but as the transformer models we use have no architecturally encoded order for the input, this order difference does not affect our discussion.

D Linear Programming Details

The max and min are non-linear, making it impossible to directly apply linear programming to our target loss function for creating the various data distributions. To circumvent this problem, we define two helper variables:

1. MaxValue: A variable that is constrained to be greater than or equal to the count of any possible combination, i.e. MaxValue $\geq \#TTC \wedge$ MaxValue $\geq \#FTC \ldots$.
2. MinValue, i.e. A variable that is constrained to be greater or equal to the count of any possible combination, i.e. MinValue $\leq \#TTC \wedge$ MinValue $\leq \#FTC \ldots$.

The target of the linear optimization is then to minimize MaxValue $-$ MinValue.

E Architectural Details

Like Strohmaier and Wimmer (2025), we base our experiments on the standard transformer model included in the PyTorch library (Paszke et al., 2017). As in Vaswani et al. (2017), we use an encoder-decoder architecture. We implemented the models using PyTorch and parallelised using the lightning-Fabric library. For training and evaluation, we used two A100 GPUs.

Like Strohmaier and Wimmer (2025), we use the Binary Cross Entropy (BCE) loss function. Since this is not a typical language modelling task, we have special cases, e.g. with more than one allowed embedded clause, where we set the target value for all allowed embedded clauses to 0.5. Also, if two attitude verbs are allowed, we set the target value for both to 1.

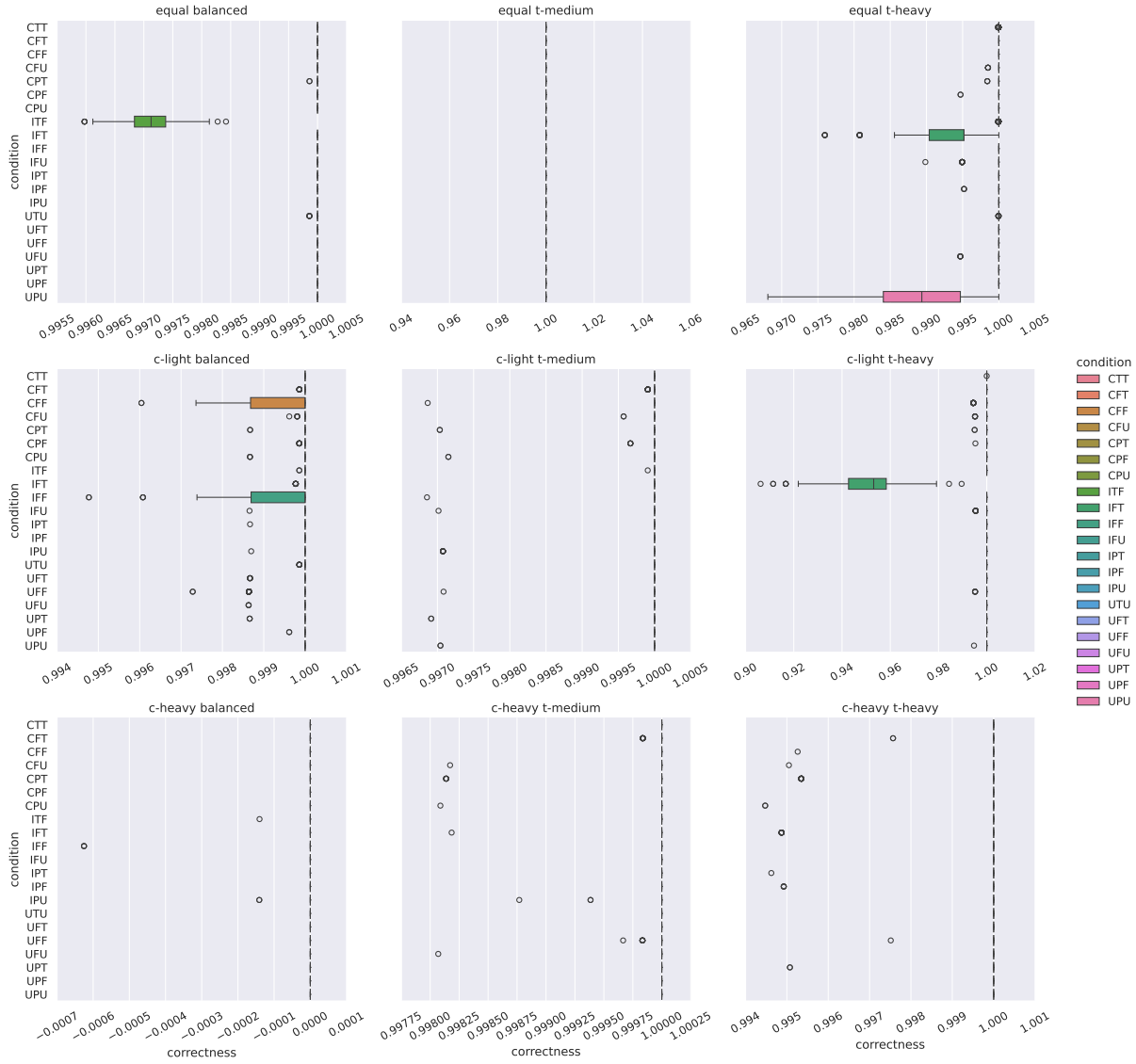
F Hyperparameter Search Details

The target of the hyperparameter search was to maximize the overall correctness of output created on the dev set. All 9×60 settings were fully explored (no pruning). The 60 settings were explored by 4 processes, which communicated via a remote mySQL database.

Name	Lower	Upper	Step Size	Log-Space
Embedding dim.	48	480	24	
Hidden dim.	48	480	24	
# Encoder layers	5	25	5	
# Decoder layer	5	25	5	
Dropout prob.	0.1	0.3	0.1	
Learning rate	1e-07	1e-03	–	✓
Epochs	5	50	1	
Batch size	8	3200	8	

Table 8: Hyperparameter settings used for search by Optuna.

G Learning Trajectory



Performance on the 21 possible conditions given in Table 1.

Significance Test We use the two-sided permutation-test implementation of SciPy (Virtanen et al., 2020) with 9999 resamples and permutation type "samples". The test statistic was the difference between the sum of comparisons in which the model performed better on factives than on contrafactives and the sum of comparisons where the reverse held.

step	f	c	nf	comb
0	9.546%	32.237%	0.027%	11.279%
200	98.280%	83.964%	99.362%	93.685%
420	99.996%	100.000%	100.000%	99.998%
640	100.000%	100.000%	100.000%	100.000%
840	100.000%	100.000%	100.000%	100.000%
1040	100.000%	100.000%	100.000%	100.000%
1260	100.000%	99.991%	100.000%	99.997%
1480	100.000%	100.000%	100.000%	100.000%
1680	100.000%	100.000%	100.000%	100.000%
1880	100.000%	100.000%	100.000%	100.000%
2100	100.000%	100.000%	100.000%	100.000%
2320	99.982%	100.000%	100.000%	99.994%
2520	100.000%	100.000%	100.000%	100.000%
2720	100.000%	100.000%	100.000%	100.000%
2920	100.000%	100.000%	100.000%	100.000%

(a) Trajectory for distribution equal/balanced

step	f	c	nf	comb
0	0.118%	0.005%	51.458%	17.745%
300	96.863%	96.503%	99.995%	97.763%
600	100.000%	100.000%	100.000%	100.000%
900	100.000%	100.000%	100.000%	100.000%
1220	100.000%	100.000%	100.000%	100.000%
1520	100.000%	100.000%	100.000%	100.000%
1820	100.000%	100.000%	100.000%	100.000%
2120	100.000%	100.000%	100.000%	100.000%
2420	100.000%	100.000%	100.000%	100.000%
2720	100.000%	100.000%	100.000%	100.000%
3020	100.000%	100.000%	100.000%	100.000%
3340	100.000%	100.000%	100.000%	100.000%
3640	100.000%	100.000%	100.000%	100.000%
3940	100.000%	100.000%	100.000%	100.000%
4220	100.000%	100.000%	100.000%	100.000%

(b) Trajectory for distribution equal/t-medium

step	f	c	nf	comb
0	6.470%	1.939%	0.000%	2.183%
2220	96.841%	97.175%	99.671%	97.962%
4420	96.786%	99.860%	99.750%	98.812%
6640	98.921%	99.888%	100.000%	99.636%
8860	99.931%	99.986%	100.000%	99.972%
11080	99.986%	100.000%	100.000%	99.995%
13280	99.995%	99.995%	100.000%	99.998%
15500	99.977%	99.986%	100.000%	99.992%
17720	99.977%	99.977%	100.000%	99.992%
19920	100.000%	100.000%	100.000%	100.000%
22140	99.977%	99.921%	99.995%	99.972%
24360	100.000%	100.000%	100.000%	100.000%
26580	100.000%	100.000%	100.000%	100.000%
28780	99.935%	99.935%	99.884%	99.939%
30980	100.000%	100.000%	100.000%	100.000%

(c) Trajectory for distribution equal/t-heavy

step	f	c	nf	comb
0	61.151%	10.916%	0.000%	36.807%
40	61.151%	10.916%	0.000%	36.807%
60	61.151%	10.916%	0.000%	36.807%
100	61.151%	21.688%	0.000%	39.146%
140	61.146%	21.688%	47.956%	49.534%
180	61.146%	21.681%	48.338%	49.618%
200	61.138%	21.695%	48.338%	49.620%
240	61.538%	38.426%	48.763%	53.573%
280	64.731%	43.045%	52.809%	57.279%
300	67.472%	47.514%	56.854%	60.692%
340	86.343%	78.463%	81.806%	83.643%
380	99.569%	99.280%	99.264%	99.448%
420	99.997%	100.000%	100.000%	99.998%
440	100.000%	100.000%	100.000%	100.000%
460	100.000%	100.000%	100.000%	100.000%

(d) Trajectory for distribution c-light/balanced

step	f	c	nf	comb
0	42.654%	5.093%	0.000%	25.158%
20	42.657%	5.093%	0.000%	25.160%
60	40.413%	5.945%	0.783%	24.185%
80	44.969%	14.543%	6.297%	29.850%
100	49.181%	18.629%	13.273%	34.640%
120	53.754%	28.557%	19.080%	40.651%
160	75.072%	56.298%	45.298%	64.562%
180	97.706%	88.000%	74.814%	90.744%
200	99.915%	92.460%	77.980%	93.664%
240	99.481%	91.894%	89.871%	95.934%
260	99.950%	92.585%	100.000%	98.420%
280	99.992%	95.760%	100.000%	99.098%
300	99.992%	99.978%	100.000%	99.995%
340	99.992%	99.978%	100.000%	99.995%

(e) Trajectory for distribution c-light/t-medium

step	f	c	nf	comb
0	4.480%	4.310%	0.089%	2.920%
20	2.050%	7.598%	0.251%	2.256%
60	9.673%	13.353%	7.377%	9.419%
80	18.697%	25.617%	17.778%	19.503%
100	43.512%	43.650%	41.817%	42.950%
140	99.023%	96.623%	99.076%	98.625%
160	99.465%	97.223%	100.000%	99.100%
200	99.502%	99.993%	100.000%	99.706%
220	99.782%	100.000%	100.000%	99.872%
240	99.691%	100.000%	100.000%	99.819%
280	99.920%	100.000%	100.000%	99.953%
300	100.000%	100.000%	100.000%	100.000%
320	100.000%	100.000%	100.000%	100.000%
360	100.000%	100.000%	100.000%	100.000%

(f) Trajectory for distribution c-light/t-heavy

step	f	c	nf	comb
0	55.907%	39.030%	0.000%	33.341%
100	51.340%	63.507%	23.683%	47.625%
200	68.717%	69.152%	48.880%	63.139%
300	95.509%	95.849%	92.343%	94.797%
420	100.000%	100.000%	100.000%	100.000%
520	100.000%	100.000%	100.000%	100.000%
620	100.000%	100.000%	100.000%	100.000%
720	100.000%	100.000%	100.000%	100.000%
820	100.000%	100.000%	100.000%	100.000%
920	100.000%	100.000%	100.000%	100.000%
1020	100.000%	100.000%	100.000%	100.000%
1140	100.000%	100.000%	100.000%	100.000%
1240	100.000%	100.000%	100.000%	100.000%
1340	100.000%	100.000%	100.000%	100.000%
1420	100.000%	100.000%	100.000%	100.000%

(g) Trajectory for distribution c-heavy/balanced

step	f	c	nf	comb
0	32.243%	6.649%	0.000%	16.257%
40	30.873%	31.429%	0.016%	28.068%
60	28.732%	32.060%	0.147%	27.316%
100	41.997%	45.298%	15.857%	40.489%
140	86.567%	87.152%	71.716%	84.168%
180	97.307%	97.474%	86.237%	94.972%
200	97.797%	97.843%	87.003%	95.516%
240	97.902%	96.719%	87.027%	95.344%
280	99.923%	99.944%	87.027%	97.469%
300	99.944%	99.978%	87.027%	97.478%
340	99.916%	99.858%	87.027%	97.441%
380	100.000%	100.000%	87.027%	97.513%
420	100.000%	99.997%	100.000%	99.998%
440	100.000%	100.000%	100.000%	100.000%
460	100.000%	100.000%	100.000%	100.000%

(h) Trajectory for distribution c-heavy/t-medium

step	f	c	nf	comb
0	1.607%	5.404%	0.048%	2.438%
60	27.054%	32.388%	25.079%	28.802%
120	90.888%	94.114%	92.854%	93.088%
200	96.647%	97.351%	97.101%	97.162%
260	97.690%	95.760%	94.115%	96.433%
320	97.869%	99.111%	94.115%	97.878%
380	98.148%	96.499%	94.115%	96.941%
440	98.316%	98.588%	94.115%	97.811%
520	99.114%	96.122%	94.115%	97.095%
580	99.187%	96.653%	94.115%	97.305%
640	99.253%	97.092%	94.173%	97.541%
700	99.865%	98.639%	95.955%	98.705%
780	100.000%	100.000%	100.000%	100.000%
840	99.993%	100.000%	100.000%	99.997%
880	99.993%	100.000%	100.000%	99.997%

(i) Trajectory for distribution c-heavy/t-heavy

Table 9: Learning Trajectory for 9 distributions (factive=f, contrafactive=c, non-factive=nf, comb=combined). "step" refers to the number of training steps taken, which is equivalent to the number of batches seen.

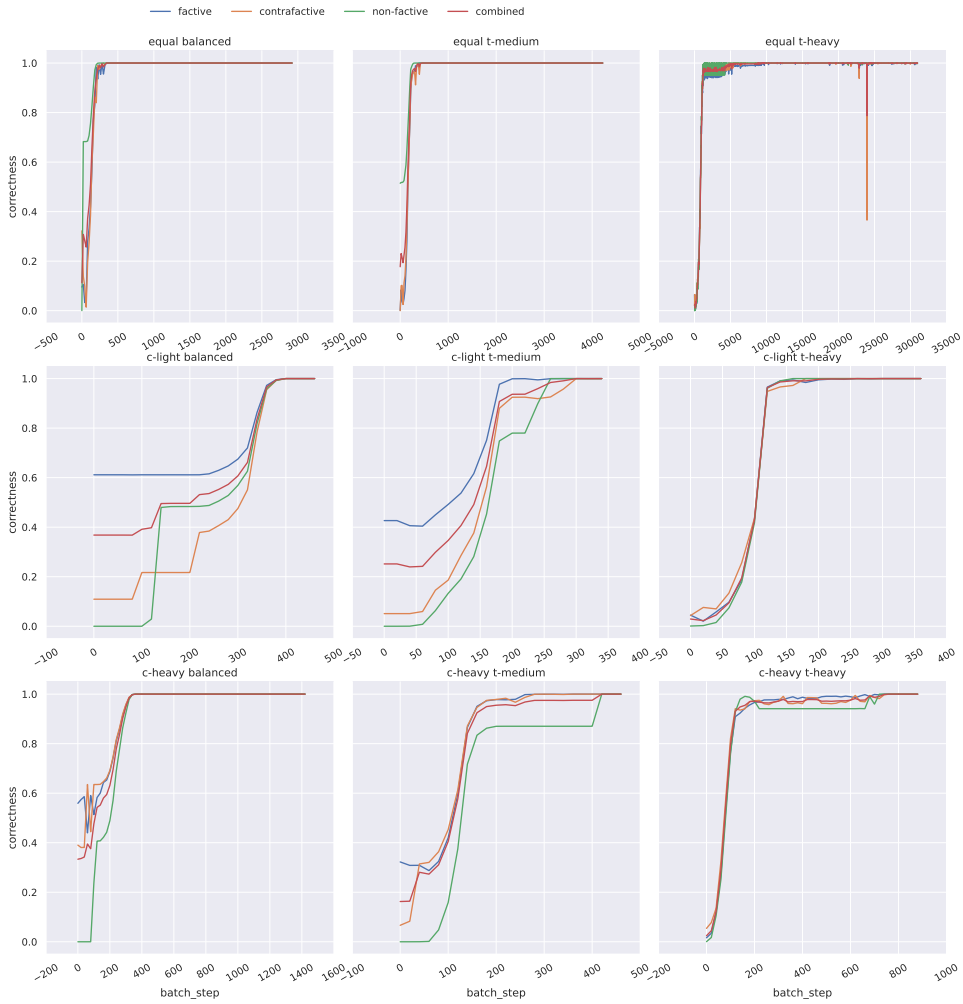
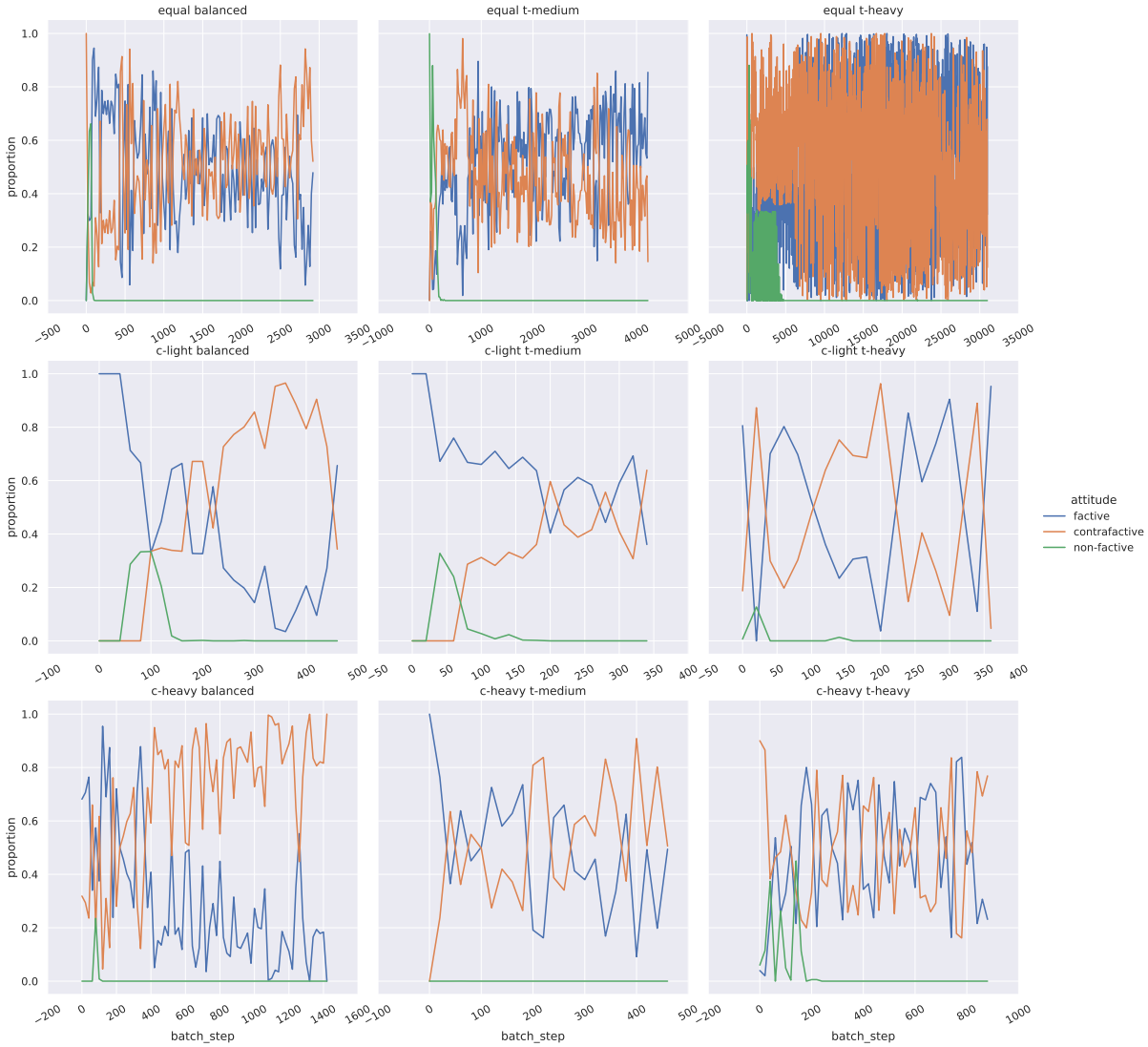


Figure 9: Changes in performance over the training for 3 attitude verbs and 9 data distributions.

H Preference where Factives and Contrafactives Both Permitted



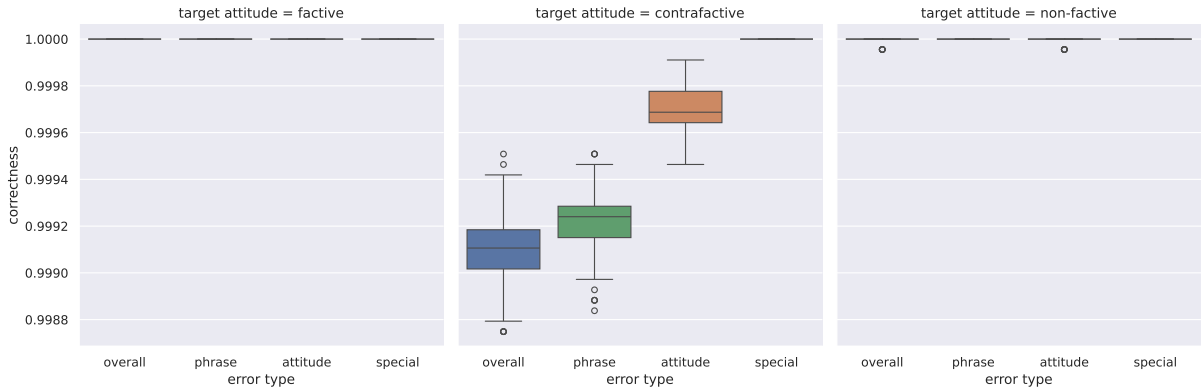
Changes over time in attitude verb preference across target distributions.

I Error Types

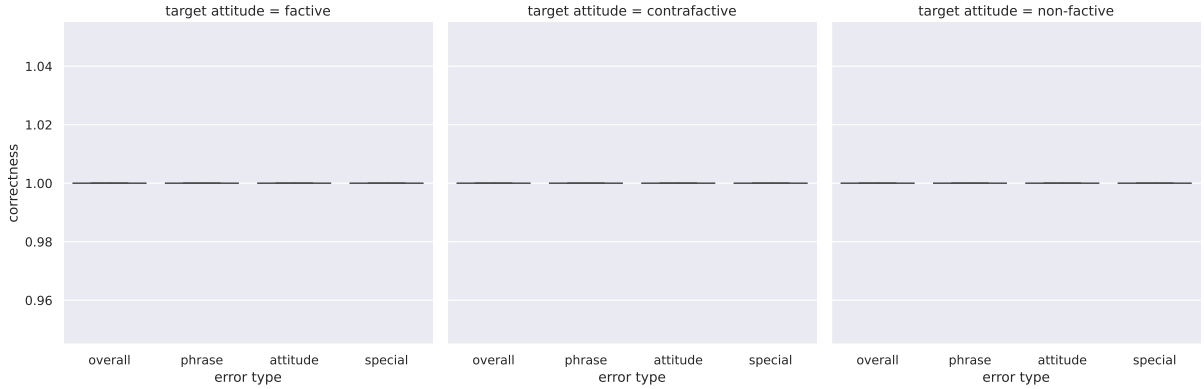
We can distinguish three parts of an output that can be responsible for an error:

1. the tokens of the embedded clause,
2. the token for the attitude verb, or
3. the special tokens

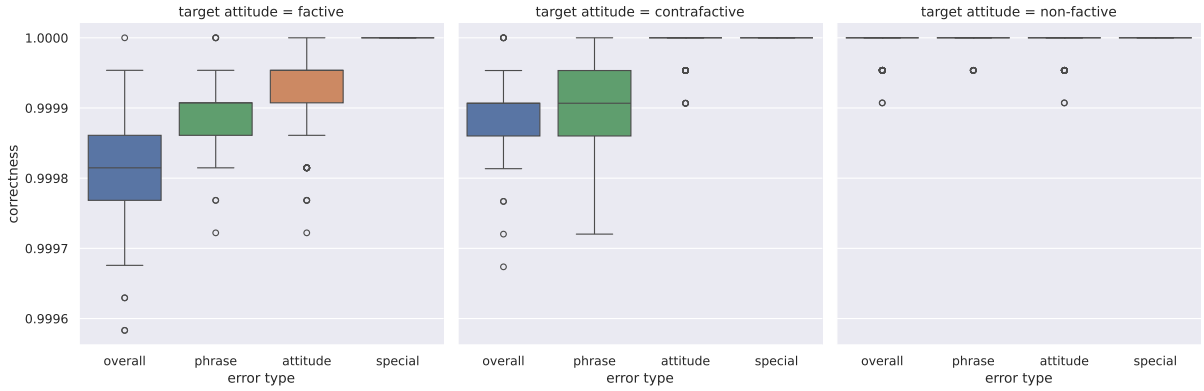
Because the vocabulary is fixed for each token position special token errors are architecturally impossible. That we find no such errors merely reflects the absence of coding errors. Notably, the worst performance on input that allows contrafactives, found in distribution equal/balanced, is primarily due to errors in the production of embedded clause tokens.



(a) Error types for allowed attitude verbs given the distribution equal/balanced.

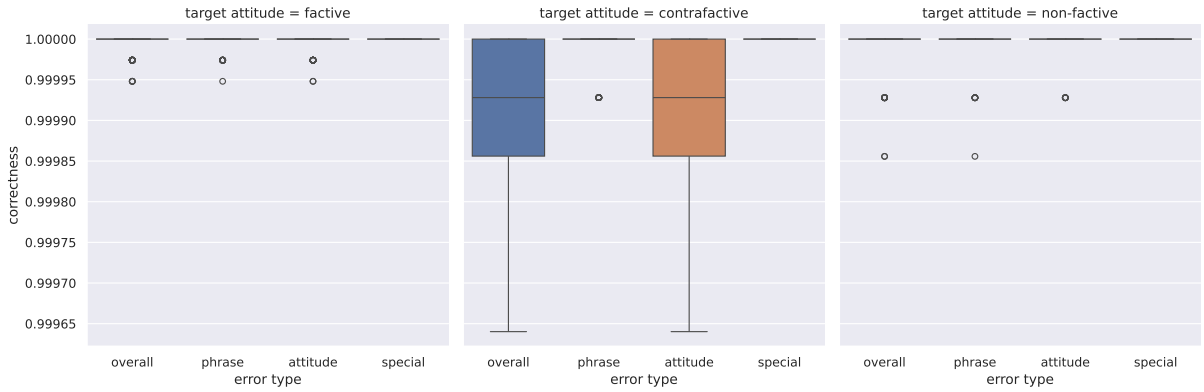


(b) Error types for allowed attitude verbs given the distribution equal/t-medium.

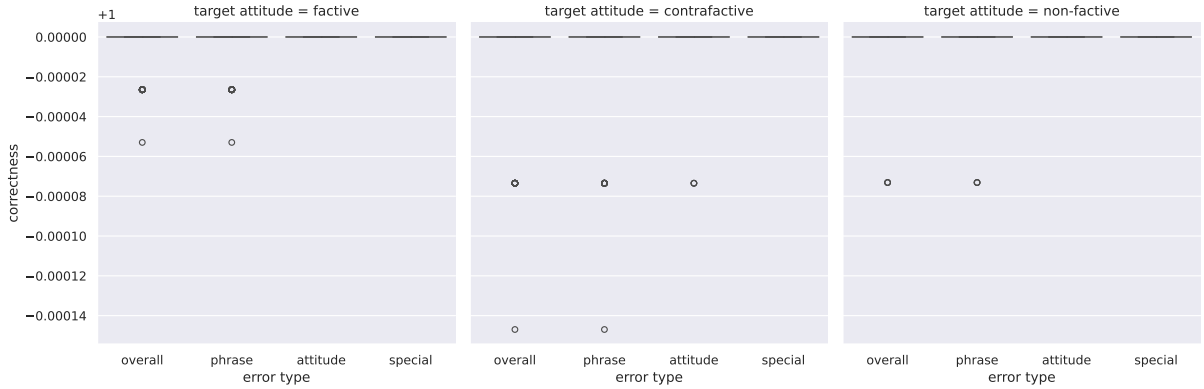


(c) Error types for allowed attitude verbs given the distribution equal/t-heavy.

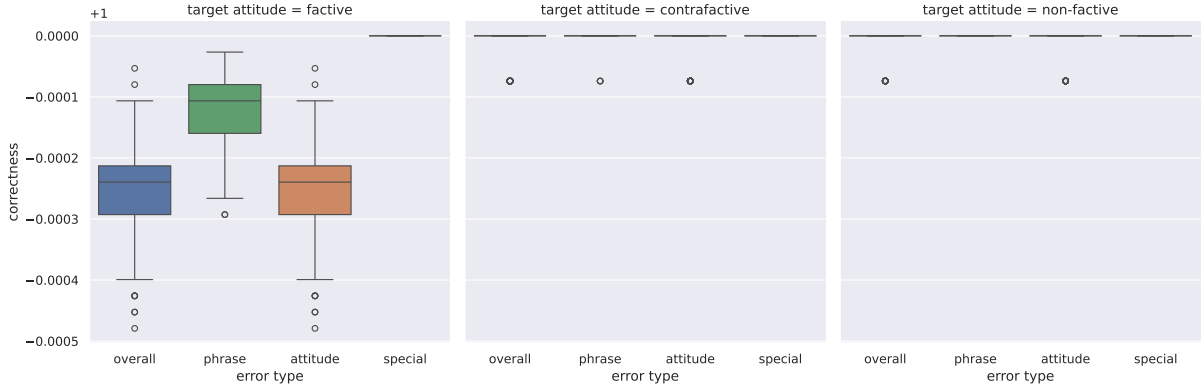
Figure 10: Error types by attitude verb given distributions that are equal.



(a) Error types for allowed attitude verbs given the distribution c-light/balanced.

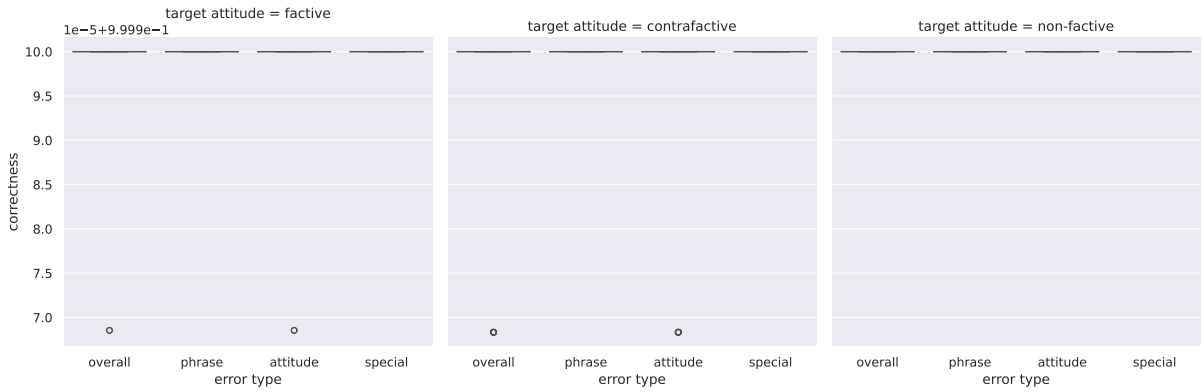


(b) Error types for allowed attitude verbs given the distribution c-light/t-medium.

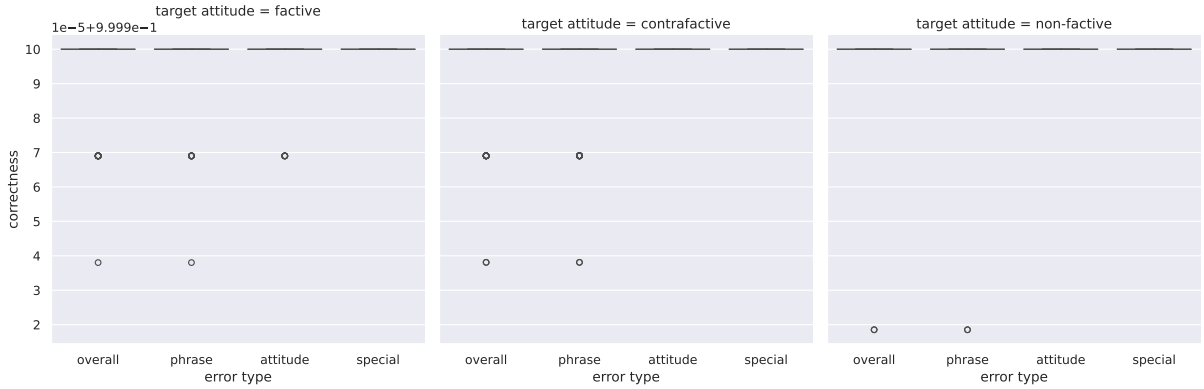


(c) Error types for allowed attitude verbs given the distribution c-light/t-heavy.

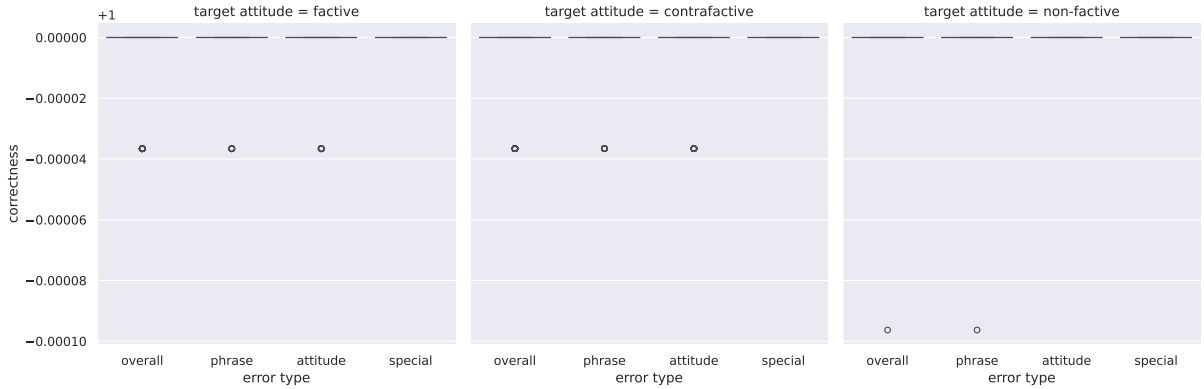
Figure 11: Error types by attitude verb given distributions that are c-light.



(a) Error types for allowed attitude verbs given the distribution c-heavy/balanced.



(b) Error types for allowed attitude verbs given the distribution c-heavy/t-medium.

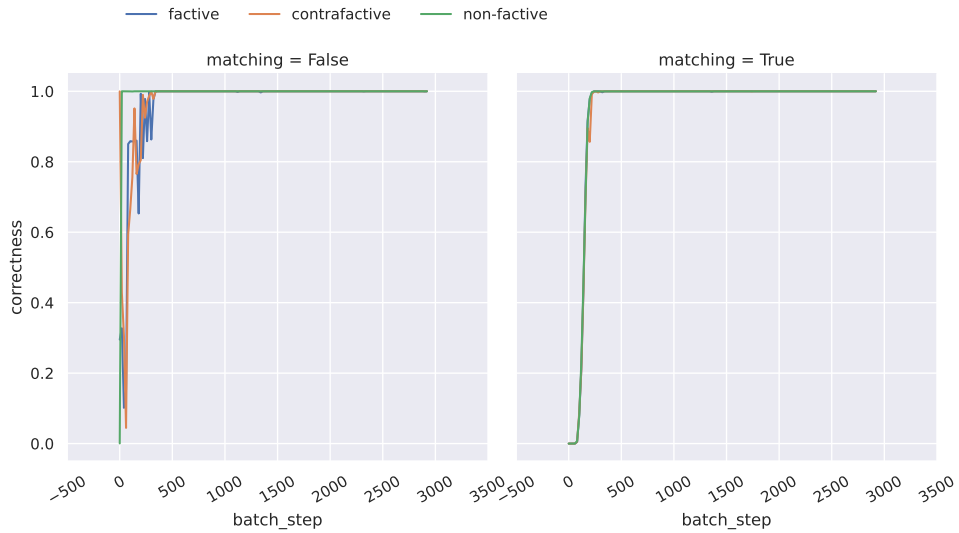


(c) Error types for allowed attitude verbs given the distribution c-heavy/t-heavy.

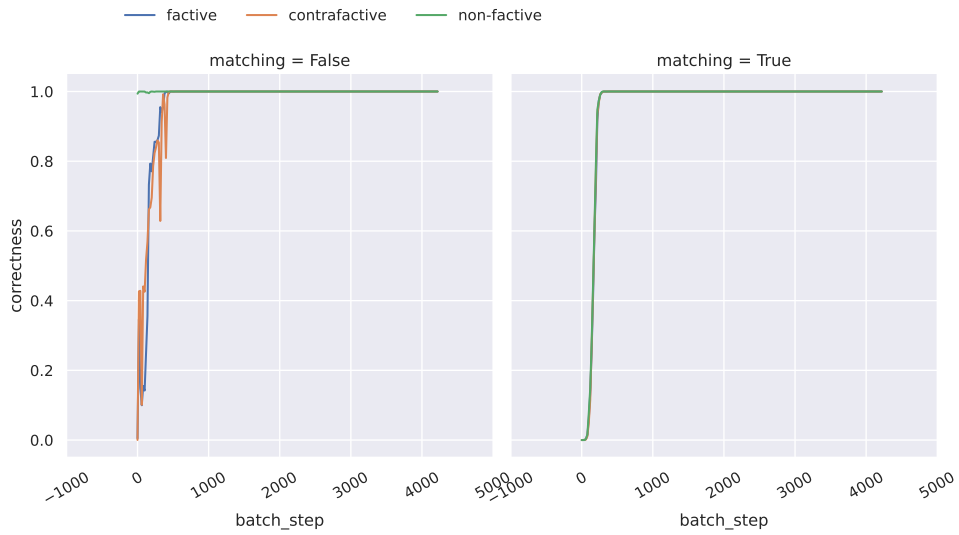
Figure 12: Error types by attitude verb given distributions that are c-heavy.

J Matching vs. Non-Matching

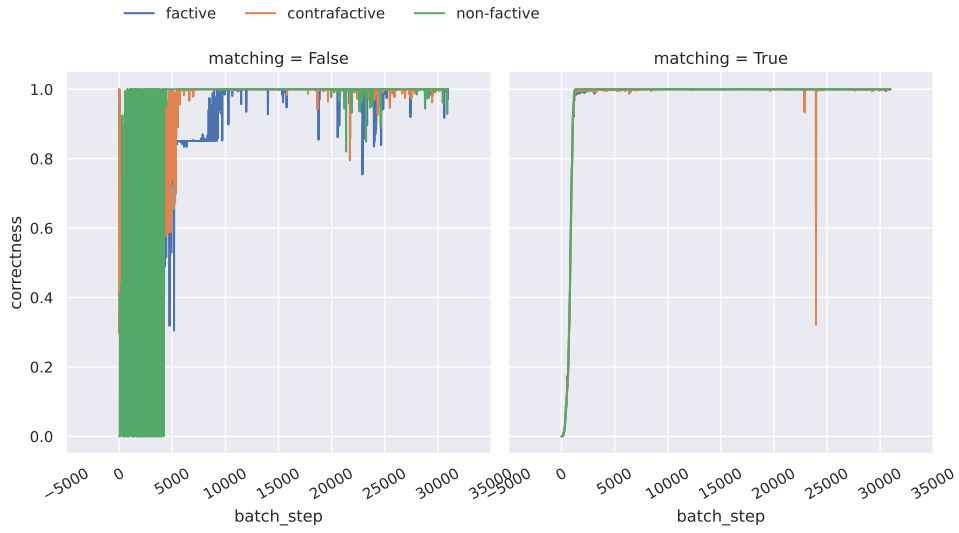
Strohmaier and Wimmer (2025) reported strong differences in performance between conditions that require matching embedded clauses and conditions that require non-matching embedded clauses (see Table 1 to see into which group each of the 21 conditions falls). We provide here the learning graphs split up by what kind of embedded clause is required for all 9 distributions. Our results replicate those of Strohmaier and Wimmer (2025) across distributions.



(a) Matching vs. non-matching performance for allowed attitude verbs given the distribution equal/balanced.

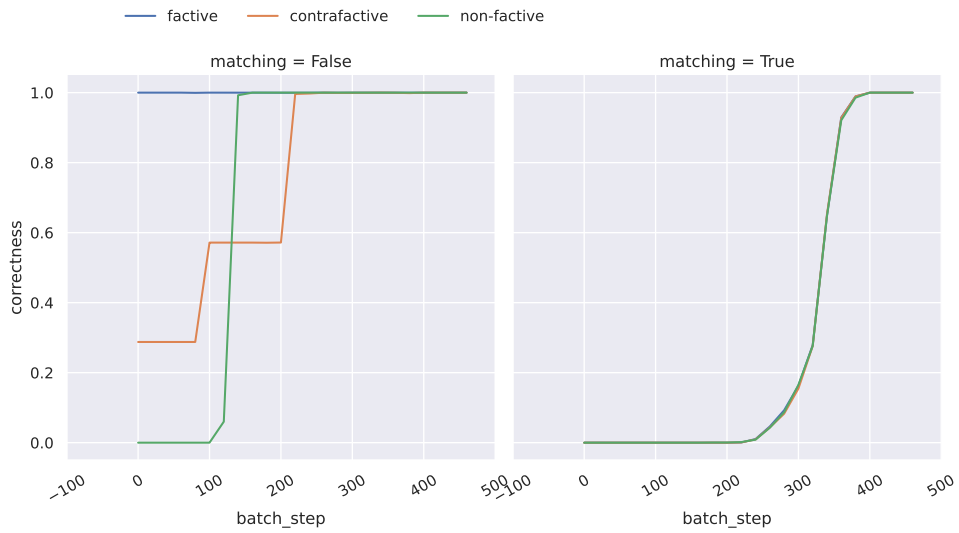


(b) Matching vs. non-matching performance for allowed attitude verbs given the distribution equal/t-medium.

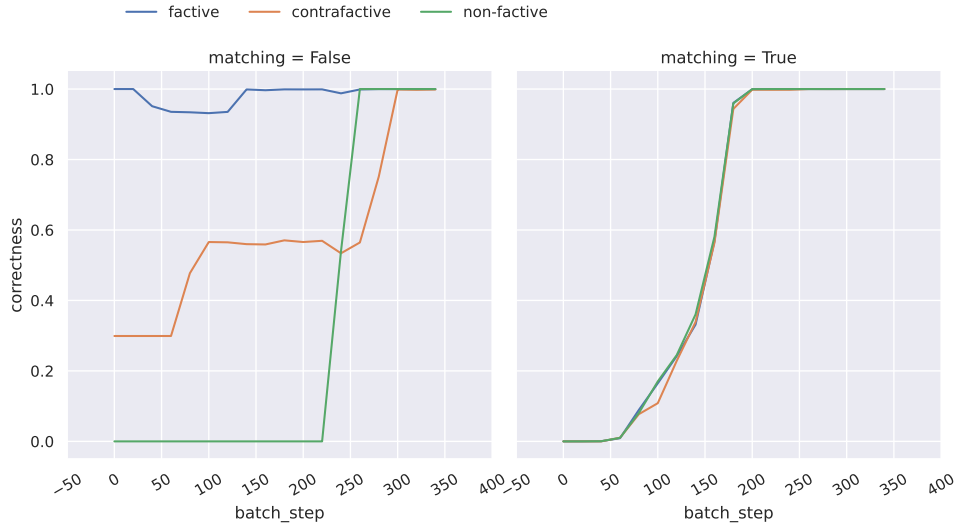


(c) Matching vs. non-matching performance for allowed attitude verbs given the distribution equal/t-heavy.

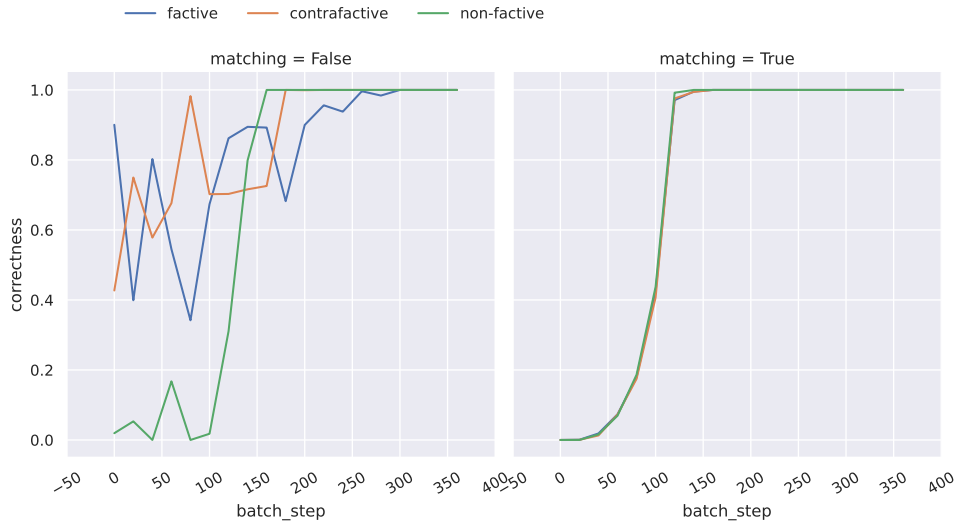
Figure 13: Matching vs. non-matching performance by attitude verb given distributions that are equal.



(a) Matching vs. non-matching performance for allowed attitude verbs given the distribution c-light/balanced.

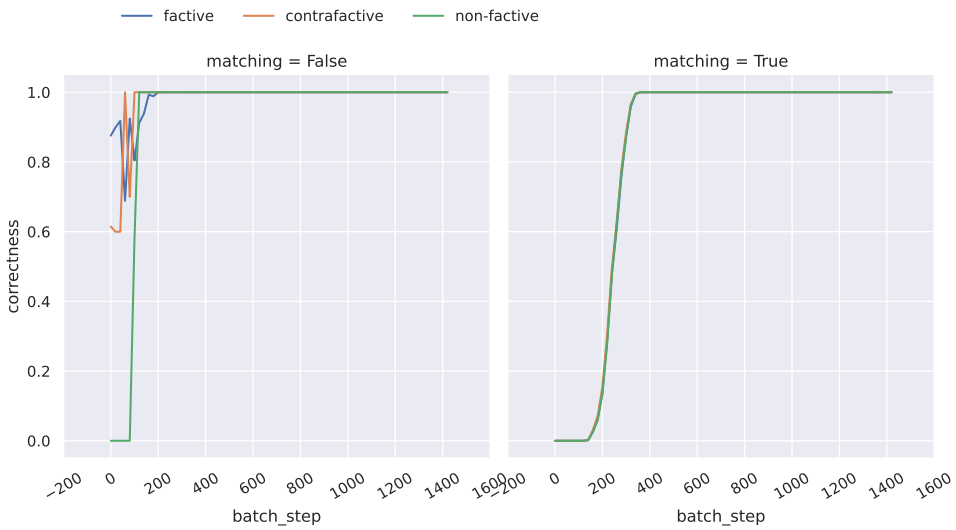


(b) Matching vs. non-matching performance for allowed attitude verbs given the distribution c-light/t-medium.

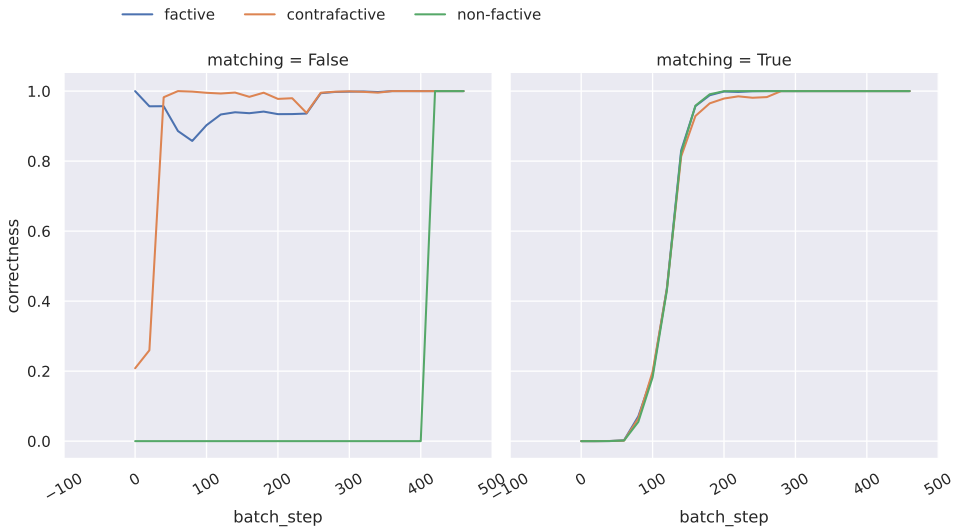


(c) Matching vs. non-matching performance for allowed attitude verbs given the distribution c-light/t-heavy.

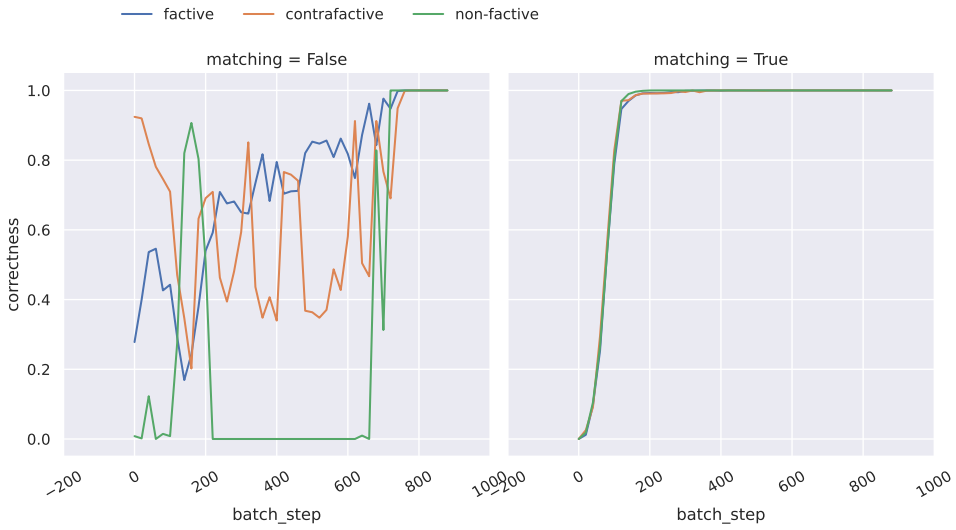
Figure 14: Matching vs. non-matching performance by attitude verb given distributions that are c-light.



(a) Matching vs. non-matching performance for allowed attitude verbs given the distribution c-heavy/balanced.



(b) Matching vs. non-matching performance for allowed attitude verbs given the distribution c-heavy/t-medium.



(c) Matching vs. non-matching performance for allowed attitude verbs given the distribution c-heavy/t-heavy.

Figure 15: Matching vs. non-matching performance by attitude verb given distributions that are c-heavy.

Grammar Engineering Meets LLMs: Development of Cantonese and Irish ParGram Treebanks

Chit-Fung Lam
University of Oxford
University of Manchester
lawrence.lam@ling-phil.ox.ac.uk

Elaine Uí Dhonnchadha
Trinity College Dublin
uidhonne@tcd.ie

Abstract

Grammar engineering requires expertise in linguistic formalism and computational implementation, particularly in parallel grammar projects that balance cross-linguistic consistency with language-specific properties. This paper presents the development of Cantonese and Irish treebanks within the Parallel Grammar (ParGram) Project, where linguistic parallelism is maintained at an abstract functional level. We also investigated the methodological potential and limitations of using multilingual LLMs to support grammar engineering, focusing on Cantonese–Irish translation and the generation of formal syntactic structures using OpenAI's `gpt-oss-120b`. The results showed that translation performance was generally unsatisfactory and unaffected by prompt language. For syntactic structure generation, the model produced some structurally meaningful outputs, but performed poorly on tasks requiring cross-linguistic abstraction. Nonetheless, LLM-generated outputs may still offer some reference value by suggesting alternative analyses and (partially) capturing predicate–argument relations. Overall, our findings highlight both the potential and limitations of using LLMs in collaborative grammar engineering, while underscoring the continued importance of expert-driven analysis and verification.

1 Introduction

Grammar engineering involves the design and computational implementation of linguistically well-motivated grammars (Duchier and Parmentier, 2015). It requires precise, rule-based systems—often within formal frameworks such as Lexical Functional Grammar (LFG) and Head-driven Phrase Structure Grammar (HPSG)—that map natural language utterances to deep representations encoding syntactic and semantic relations (Bender, 2008; Forst and King, 2023; Zamaraeva et al., 2022). The Parallel Grammar (ParGram) Project is a large-scale, ongoing effort aimed at developing parallel LFG grammars for typologically diverse languages (Sulger et al., 2013; Butt et al., 1999). ParGram grammars are implemented in the Xerox Linguistic Environment (XLE; Crouch et al. 2011). Parsing with these grammars yields LFG's c-structure and f-structure representations (Section 2.2). The c-structure captures constituency, word order, and part-of-speech information, while the f-structure encodes grammatical functions (e.g., SUBJ, OBJ) and associated features such as tense, aspect, number, and gender (Dalrymple et al., 2019). ParGramBank comprises of the family of ParGram treebanks obtained by applying LFG/XLE grammars to a parallel corpus translated across languages (Rosén, 2023). These treebanks maintain a level of parallelism in the f-structure, achieved via a shared inventory of grammatical features (King, 2004), consistent with LFG's aim of capturing language universals at an abstract functional level (Kaplan and Bresnan, 1982; Bresnan et al., 2016).

Cantonese and Irish are under-resourced languages, which are typologically different from each other and most well-documented languages (McCrae and Doyle, 2019; Xiang et al., 2024). Their contrasting grammatical properties provide a challenging test case for maintaining grammar parallelism within ParGram while encoding language-specific differences. In Section 2, we describe some of the challenges encountered in representing language-specific features while maintaining grammar parallelism across languages.

Grammar engineering is inherently knowledge- and labour-intensive, requiring expertise in both linguistic theory and computational implementation. With the advent of large language models (LLMs), there has been growing interest in whether such models can assist grammar development (Spencer and Kongborrirak, 2025) and whether they exhibit the metalinguistic capabil-

Third Workshop on the Bridges and Gaps between Formal and Computational Linguistics, pages 121–134

ities needed to produce deep linguistic analyses, including formal syntactic structures (Begu et al., 2025; Murphy et al., 2025; Kennedy, 2025). Within the ParGram framework—particularly in the construction of ParGramBank—grammar engineers face, among others, two common tasks: (i) producing reliable translations across typologically diverse languages to build parallel corpora; (ii) specifying appropriate syntactic representations for the translated sentences. These tasks are crucial for evaluating whether existing grammars can successfully parse the data and for identifying where new constraints or refinements are required. Given that LLMs may serve as assistants in translation and structural analysis, this paper investigates the extent to which they can support grammar engineering. Specifically, we evaluate whether a multilingual LLM can provide useful cross-linguistic translations and LFG-style analyses for Cantonese and Irish in the ParGramBank setting. Our goal is not to evaluate the linguistic competence of LLMs in general, but to assess their practical usefulness as tools that may assist grammar engineers in parallel grammar development.

This paper is structured as follows. Section 2 describes the Cantonese and Irish ParGram treebanks, including their design, progress, and challenges, alongside some selected linguistic phenomena. Section 3 presents the aims, methodology, and results of evaluating a multilingual LLM (OpenAI's `gpt-oss-120b`) with respect to its potential to support grammar engineering tasks using Cantonese and Irish ParGramBank data.

2 Cantonese and Irish ParGramBank

This section presents the Cantonese and Irish ParGram treebanks, outlining key design challenges, illustrated with selected examples from the first 50 sentences of ParGramBank. These sentences are included in Appendix A.

2.1 ParGram Treebank Development

ParGram treebanks are pure parsebanks in that the c-structures and f-structures they contain are generated automatically by parsing sentences with their respective LFG/XLE grammars (Rosén, 2023). No ad hoc post-parsing corrections or manual annotations are permitted. As such, ParGram treebanks differ from many traditional treebanks, which often incorporate varying degrees of manual correction or post hoc adjustment (Rosén, 2023). A key advantage of this design is that the treebank remains fully synchronized with its underlying grammar: since all c- and f-structures are produced by the same grammar, inconsistencies between analyses are avoided. However, this tight coupling also entails a trade-off, as any error in a c- or f-structure cannot be corrected directly in the treebank but instead requires revision of the grammar itself, a process that can be time-consuming. To support ongoing development and facilitate cross-linguistic consistency, the ParGram community has established a dedicated forum for discussion. Since 2023, monthly ParGram f-structure comparison meetings bring together LFG/XLE grammar engineers to address theoretical and implementation issues.[1] Grammar development is iterative: errors are typically identified through parsing failures, unexpected analyses, and comparison across ParGram languages. Regression testing using testsuite sentences is an intrinsic part of the XLE grammar development cycle.[2]

ParGramBank originally comprised parallel treebanks for twelve languages from six families, based on a shared corpus of 101 sentences. The corpus, initially derived from a tractor manual in English, French, and German, was expanded to include missing core syntactic constructions (Butt et al., 1999). The first 50 sentences, which are reported on in this paper, cover a wide range of phenomena, including transitivity, unaccusative–unergative distinctions, clause types, voice alternations, dative and double object constructions, complementation, control and raising, relative clauses, negation, phrasal verbs, prepositional phrases, copula constructions, reflexives, reciprocals, modification, expletives, existentials, and comparatives. Following Lehmann et al. (1997) and Bender et al. (2011), ParGram treats broad grammatical coverage as a central goal of grammar engineering (Sulger et al., 2013).

2.2 Cantonese and Irish: c-structure variation and f-structure parallelism

Cantonese and Irish are typologically distinct: Cantonese, a Sino-Tibetan language, is predominantly SVO with little morphological inflection, whereas Irish, an Indo-European language (Celtic

[1] https://sites.google.com/view/pargram-meetings

[2] https://ling.sprachwiss.uni-konstanz.de/pages/xle/doc/xle.html#TS.regression

branch), is VSO with comparatively rich inflectional morphology. Examples (1) and (2) give translation equivalents of *The driver starts the tractor* (ParGramBank, sentence 1). As shown in Figures 1 and 2, although the c-structures differ in reflecting surface order, the f-structures encode the same predicate–argument relations, specifically the verbal subcategorization for SUBJ and OBJ.[3] This demonstrates how parallelism can be maintained at the f-structural level.

(1) 個 司機 啓動 拖拉機
CL driver start tractor
'The driver starts the tractor.' (Cantonese)

(2) Tosaíonn an tiománaí an tarracóir
starts the driver the tractor
'The driver starts the tractor.' (Irish)

Besides maintaining cross-linguistic parallelism, the f-structures in Figure 2 also encode language-specific differences between Cantonese and Irish in their syntax. For example, Irish exhibits tense inflection, whereas Cantonese lacks overt tense marking but encodes aspectual distinctions, including progressive, perfective, and experiential aspects (Matthews and Yip, 2013). Following King's (2004) ParGram feature conventions, the Cantonese treebank employs the feature TNS-ASP as a cross-linguistically shared feature. However, its values do not include tense specification, reflecting the properties of Cantonese. Ongoing work expands the feature–value inventory for aspect marking (Matthews and Yip, 2013). A similar contrast arises in nominal domains: Irish noun phrases are morphologically case-marked, whereas Cantonese noun phrases are not. In addition, Cantonese is a classifier language, where classifiers and measure words occur in complementary distribution within noun phrases. This is captured by the CLASS-MEASURE feature and its associated TYPE value in the Cantonese treebank, while no corresponding feature is required for Irish.

2.3 Design, progress, and challenges of Cantonese and Irish ParGramBank

Both the Cantonese and Irish ParGram treebanks are under active expansion, alongside the ongoing development of their LFG/XLE grammars. This paper primarily targets the first 50 sentences in ParGramBank (Section 2.1). A recurring challenge in their co-development is maintaining cross-linguistic parallelism, particularly in complex constructions where word order differences (Section 2.2) become more prominent. Examples (3) and (4) illustrate an object relative clause embedded within a copula structure (ParGramBank, sentence 19).[4] In Cantonese, the relative clause appears pre-nominal with a relativizer, whereas in Irish it is post-nominal, and differences in SV vs. VS word order arise both in the copula construction and within the embedded clause. Despite these surface differences, the resulting f-structures (Figure 3) exhibit a high degree of parallelism, achieved by representing (i) the copular complement as a PREDLINK function predicating over the SUBJ (Butt et al., 1999), (ii) the relative clause as an embedded ADJUNCT within the SUBJ, and (iii) the object gap as an empty pronoun (*null-pro*) present in f-structure but absent from c-structure. These examples show how ParGram engineers maintain f-structural parallelism while accommodating language-specific properties, and highlight the importance of incorporating linguistic insights, such as gap-based relative clause strategies (Tallerman, 2020, ch 8), without compromising computational robustness.

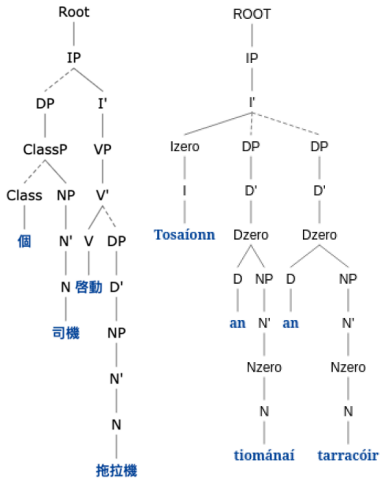


Figure 1: Cantonese and Irish c-structures for examples (1) and (2).

[3]The c- and f-structures presented in this paper were generated using the online INESS interface (Rosén et al., 2012), following the upload of the Cantonese and Irish LFG/XLE grammars to the server with assistance from Paul Meurer.

[4]An alternative Irish translation which may sound more natural to some speakers is *Tá dath dearg ar an tarracóir a cheannaigh an feirmeoir* 'There is a red color on the tractor that the farmer bought.'

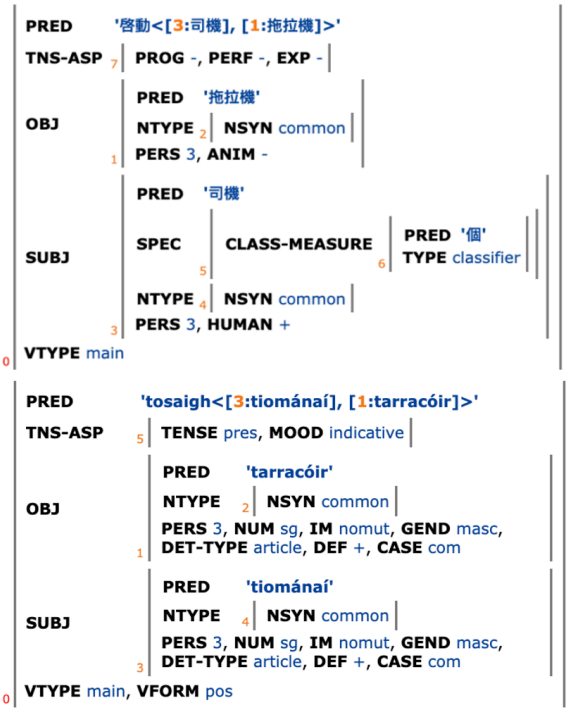


Figure 2: Cantonese and Irish f-structures for examples (1) and (2).

(3) 個 農夫 買 嘅 拖拉機 係 紅色-嘅
CL farmer buy REL tractor be red-ADJ
'The tractor that the farmer bought is red.' (Cantonese)

(4) Tá an tarracóir a cheannaigh an feirmeoir dearg
is the tractor that bought the farmer red
'The tractor that the farmer bought is red.' (Irish)

Another challenge concerns the translation of ParGramBank sentences. For example, in Irish, while the direct translation in (2) is grammatical, it may sound unnatural to some speakers because it does not capture the causative meaning of the sentence, namely *the driver causes the tractor to start*, rather than *the driver starts or begins an action themselves*. The alternative translation in (5) more accurately reflects this causative meaning.

Given that the ParGramBank family is constructed from translations of a parallel corpus, ensuring faithful and fluent translation is crucial. Poor translations may obscure underlying predicate–argument relations and complicate the establishment of cross-linguistic parallelism. In Section 3, we use these grammar-engineering tasks—translation and the construction of parallel

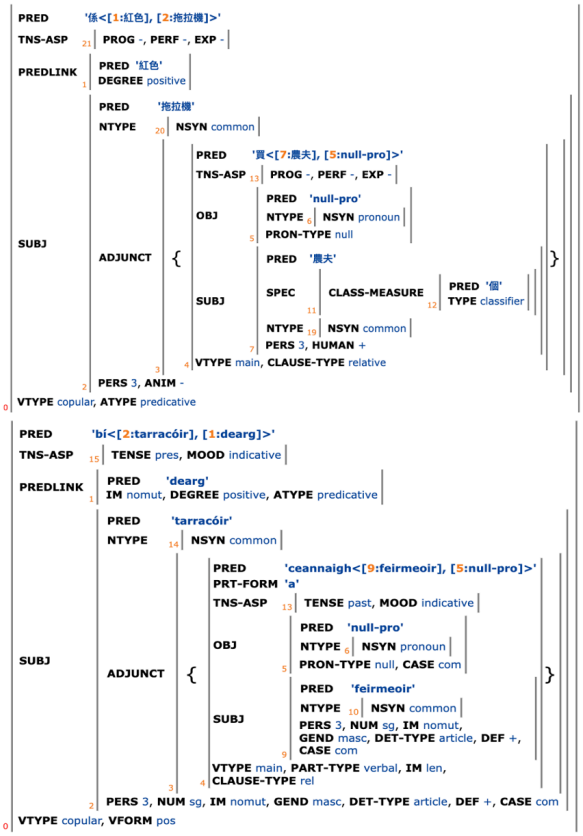


Figure 3: Cantonese and Irish f-structures for examples (3) and (4).

syntactic representations—as a case study for investigating the methodological potential and limitations of LLM-assisted grammar engineering.

(5) Déanann an tiománaí an tarracóir a thosú
makes the driver the tractor to start
'The driver makes the tractor start.'

3 Probing LLMs: Cantonese–Irish translation and LFG analysis

3.1 Aim and objectives

In this section, we investigate the methodological potential and limitations of using multilingual LLMs to support grammar engineering tasks in three areas:

1. Producing f-structures that can serve as reference points for grammar development.
2. Capturing cross-linguistic similarities between Cantonese and Irish in f-structural formats to examine grammar parallelism.

3. Generating cross-linguistic translations between Cantonese and Irish to support the construction of parallel treebanks.

No previous work has investigated translation between Cantonese and Irish, both of which are under-resourced languages (McCrae and Doyle, 2019; Xiang et al., 2024). This study represents the first examination of this language pair.

3.2 Methodology

3.2.1 Tasks and prompts

We evaluated the `gpt-oss-120b` model (OpenAI) using a zero-shot prompting setup. It is one of two open-weight models released by OpenAI, alongside `gpt-oss-20b` (OpenAI et al., 2025), and has been tested on a range of NLP benchmarks (OpenAI et al., 2025; Bi et al., 2025). We selected `gpt-oss-120b` as a representative multilingual open-weight model and use it as a case study for investigating the methodological potential and limitations of LLM-assisted grammar engineering. The goal of this study is therefore to examine the kinds of support, errors, and limitations that may arise when a contemporary multilingual LLM is incorporated into a parallel grammar-engineering workflow. Inference was performed via Hugging Face's `InferenceClient` API with temperature set to zero. We conducted six evaluation tasks for Cantonese and Irish:

- Tasks 1 and 2: f-structure generation (English prompt)
- Tasks 3 and 4: translation combined with f-structure generation (English prompt)
- Tasks 5 and 6: translation only (Cantonese/Irish prompt)

In Task 1, the model generated an f-structure for each of the first 50 Cantonese sentences in ParGramBank, using the following prompt format:

```
You are an expert in Lexical Functional Grammar (LFG) and Cantonese. Provide one f-structure for the Cantonese sentence in triple single quotes '''...'''. Make sure the f-structure is in the attribute-value matrix format.
```

Task 2 followed the same procedure for Irish: the model was asked to generate an f-structure for each of the 50 Irish sentences. As a baseline, we evaluated the model on the first 50 English ParGramBank sentences (Butt et al., 1999; Sulger et al., 2013) using the same prompt format.

In Task 3, the model first translated each of the 50 Cantonese sentences into Irish and then generated an f-structure intended to reflect the shared structural properties of the source and translated sentences, using the following prompt format:

```
You are a linguist specializing in Lexical-Functional Grammar and a language expert in Cantonese and Irish. Your task is to first translate the Cantonese sentence in triple single quotes '''...''' into Irish; your translation should be both faithful and fluent. Then, provide one f-structure to capture the linguistic similarities between Cantonese and Irish with regard to this translated sentence.
```

Task 4 mirrored Task 3, but in the opposite direction: the model was asked to translate each of the 50 Irish sentences into Cantonese and then generate a corresponding f-structure.

Tasks 5 and 6 replicate the translation components of Tasks 3 and 4, respectively, but without f-structure generation: the model was asked to translate Cantonese sentences into Irish (Task 5) and Irish sentences into Cantonese (Task 6), with prompts formulated entirely in the respective source languages. These tasks were motivated by previous work (Cao et al., 2023), which suggests that varying the prompt language may affect model performance (cf. Hautli-Janisz et al., 2025).

3.2.2 Evaluation criteria

For tasks involving LLM-generated f-structures (Tasks 1–4), the Cantonese and Irish ParGram treebanks (Section 2) served as the gold standard for evaluation, while allowing for alternative linguistically valid analyses. We therefore adopted a four-point scale (**Excellent**, **Good**, **Fair**, **Bad**) to assess how well outputs conformed to LFG principles, prioritizing correct grammatical functions (GFs) and predicate–argument structure, with non-GF features (e.g., tense, aspect, person, number, and gender) considered secondarily. In brief, **Excellent** denotes a fully correct analysis, **Good** a structurally correct analysis with minor feature-level errors, **Fair** a partially correct analysis in which the core construction remains recognizable, and **Bad** an analysis that fails to capture the intended construction. Full details of the rating scheme are provided in Appendix B. F-structure evaluation

was performed by the authors, who are Cantonese and Irish ParGram grammar engineers. The Cantonese and Irish ParGram treebanks served as the evaluation gold standards, and the treebanks have been discussed and refined through ongoing ParGram f-structure comparison meetings.[5] Evaluation of generated f-structures was performed entirely through expert manual assessment; no automated well-formedness checking or parser-based validation was applied.

For the translation tasks (Tasks 3–6), translation quality was evaluated independently in terms of faithfulness and fluency, as specified in the prompt, since these are not reliably captured by surface-form metrics such as BLEU. Given the absence of standard Cantonese–Irish reference translations, automatic evaluation is difficult to apply meaningfully. **Faithfulness** was defined as preservation of the core meaning of the source sentence, while **fluency** was assessed in terms of morphological and syntactic well-formedness as well as idiomaticity for native speakers of the target language. For each translation, faithfulness and fluency were judged separately using binary (Y/N) decisions. Each translation was evaluated independently by two annotators with expertise in the relevant languages. Disagreements were resolved through discussion. Inter-annotator agreement ranged from moderate to almost perfect (Cohen's $\kappa = 0.45$–0.92), as shown in Table 1. A translation was considered successful only if it was both faithful and fluent. We report faithfulness, fluency, and overall success rates separately.

Traslation Task	Criterion	κ
Irish→Cantonese (English prompt)	Faithfulness	0.92
Irish→Cantonese (English prompt)	Fluency	0.62
Irish→Cantonese (Irish prompt)	Faithfulness	0.84
Irish→Cantonese (Irish prompt)	Fluency	0.71
Cantonese→Irish (English prompt)	Faithfulness	0.63
Cantonese→Irish (English prompt)	Fluency	0.75
Cantonese→Irish (Cantonese prompt)	Faithfulness	0.45
Cantonese→Irish (Cantonese prompt)	Fluency	0.63

Table 1: Inter-annotator agreement for translation evaluation, measured using Cohen's κ.

3.3 Results

The results of the translation tasks, based on the final adjudicated ratings, are presented in Table 2. We varied the prompt language with respect to both the source and target languages. For example, in the Cantonese-to-Irish tasks, prompts were provided in either English or Cantonese, simulating a scenario in which a Cantonese corpus is available and grammar engineers, who may not be familiar with Irish, translate Cantonese sentences into Irish to support the development of an Irish component. Conversely, for Irish-to-Cantonese translation, prompts were given in either English or Irish, reflecting a complementary scenario in which an Irish corpus is available and engineers translate Irish sentences into Cantonese to support the development of a Cantonese component.

For Irish-to-Cantonese translation, faithfulness scores (44–50%) were consistently higher than fluency scores (32–34%). Requiring translations to satisfy both criteria substantially reduced overall success rates (22–24%), suggesting that translations often preserved some aspects of meaning without producing fully natural target-language output.

The results for Cantonese-to-Irish translation showed the opposite trend, with faithfulness scores 8–12%, fluency scores 18–22%, and both criteria 6–8%. Higher fluency than faithfulness may reflect the lack of explicit tense and gender information in some of the Cantonese input, resulting in fluent but non-faithful Irish translations. Data sparsity issues associated with the more inflected language, and the models' propensity to invent words contributed to the lower faithfulness and fluency. However, if we ignore hallucinated words the fluency scores increase to 36-40%. The poorer overall results for Irish suggest that the LLM's training data may have contained less Irish data than Cantonese data.

For each source–target language pair, we conducted post hoc McNemar's tests to assess whether prompt language had a significant effect on overall translation success (i.e., translations judged both faithful and fluent). For Irish-to-Cantonese translation, an exact McNemar test on translations judged both faithful and fluent showed no significant difference between the English and Irish prompt conditions ($p = 1.00$). Likewise, for Cantonese-to-Irish translation, no significant difference was found between the English and Cantonese prompt conditions ($p = 1.00$). In other words, prompt language did not significantly affect translation performance for this language pair for our model. This finding contrasts with previous work such as Cao et al. (2023), but is consistent with Hautli-Janisz et al. (2025). Overall, trans-

[5] https://sites.google.com/view/pargram-meetings

	Faithful	Fluent	Both
Irish-to-Cantonese (English prompt)	50%	32%	22%
Irish-to-Cantonese (Irish prompt)	44%	34%	24%
Cantonese-to-Irish (English prompt)	12%	18%	6%
Cantonese-to-Irish (Cantonese prompt)	8%	22%	8%

Table 2: Percentages of translations judged faithful, fluent, and both faithful and fluent across translation tasks.

lation performance was unsatisfactory. We discuss recurring qualitative issues in Section 3.4.

Table 3 presents the evaluation of LLM-generated f-structures for Tasks 1–4, including an English baseline. The LLM exhibited limited but non-negligible performance in generating LFG-style f-structures. The model performed best on English, with 20% of outputs rated as *Excellent* and 26% as *Good*, and a further 48% classified as *Fair*. For Cantonese (Task 1), 22% of outputs were rated as *Excellent* and 12% as *Good*, yielding 34% fully correct structural analyses. A further 40% were classified as *Fair*, indicating recognizable structures despite errors in GF assignment or subcategorization, while 26% were rated as *Bad*. For Irish (Task 2), only 2% of outputs were rated as *Excellent*, while 18% were *Good*, giving a total of 20% structurally correct analyses. As in Task 1, a substantial proportion (46%) fell into the *Fair* category, suggesting that the model often captured the general predicate–argument relations despite errors in GFs or feature specification; 34% were rated as *Bad*. Overall, while fully accurate analyses were relatively infrequent, the model captured at least some predicate–argument relations in around 70% of cases in both tasks.

To compare f-structure generation across English, Cantonese, and Irish, we coded the evaluation categories ordinally (*Excellent* = 4, *Good* = 3, *Fair* = 2, *Bad* = 1) and conducted a Friedman test, which showed a significant difference across conditions ($\chi^2(2) = 9.49, p = .009$). Pairwise Wilcoxon signed-rank tests indicated no significant difference between the English baseline and Cantonese (Task 1) ($p = .176$), but showed that both English ($Z = -3.84, p < .001, r = -.66$) and Cantonese ($Z = -2.16, p = .031, r = -.36$) significantly outperformed Irish (Task 2).

The lower ratings for Irish may be attributed to typological and data-related factors: its richer inflectional morphology requires more feature–value pairs in f-structure, increasing analytical complexity and making consistent specification more difficult, whereas Cantonese has minimal inflectional morphology, reducing this burden: once GFs are correctly identified, it is more likely for Cantonese analysis to receive the *Excellent* rating. The English baseline patterns more closely with Cantonese than with Irish, suggesting that morphologically simpler systems are handled more robustly by the model. In addition, the model may have been exposed to more Chinese than Irish data, although the extent to which this includes Cantonese is unclear.[6] Word order differences may also contribute: Cantonese and English are predominantly SVO, whereas Irish is VSO, a pattern that may be less well represented in the model's training data.

Performance on shared f-structure generation was weaker. In Task 3, 80% of sentences were rated as either *Fair* or *Bad*, with 44% classified as *Bad*. In Task 4, only 4% were rated as *Excellent*, while the majority fell into the *Fair* (50%) and *Bad* (38%) categories. A Wilcoxon signed-rank test over the four rating levels (*Excellent*/*Good*/*Fair*/*Bad*) showed that Task 1 received significantly higher ratings than Task 4 ($Z = -3.23, p = .001, r = -.65$). However, no statistically significant difference was found between Tasks 2 and 3 ($p = .991$). The additional requirement to abstract over cross-linguistic similarities appears to pose a greater challenge for the model.

3.4 Discussion: LLM-generated translation

Several recurring issues were observed in the LLM-generated translations. One prominent pattern is the intrusion of Mandarin forms in outputs intended to be Cantonese, affecting translation fluency. For example, in 我妹妹係一位好棒嘅老師 from Irish *Is múinteoir iontach í mo dheirfiúr* ('My sister is a great teacher'), the word 好棒 'great' is Mandarin rather than Cantonese; a more natural Cantonese equivalent would be 好叻. Similarly, in 佢哋為佢哋的女兒感到自豪 from Irish *Tá siad bródúil as a n-iníon* ('They are proud of their daughter'), the expression 的女兒 ('(someone's)

[6] See https://cameronrwolfe.substack.com/p/gpt-oss

	Excellent	Good	Fair	Bad
English f-structures (Baseline)	20%	26%	48%	6%
Cantonese f-structures (Task 1)	22%	12%	40%	26%
Irish f-structures (Task 2)	2%	18%	46%	34%
Cantonese–Irish shared f-structures (Task 3)	8%	12%	36%	44%
Irish–Cantonese shared f-structures (Task 4)	4%	8%	50%	38%

Table 3: Percentages of f-structures by evaluation category (Excellent, Good, Fair, Bad) in each task.

daughter') is Mandarin, whereas Cantonese would typically use 個女 '(classifier) daughter'. Another example is the use of the Mandarin object-marking construction 把, where Cantonese would instead employ 將. These cases indicate that, despite being prompted for Cantonese output, the model often produced Mandarin lexical and grammatical forms, suggesting difficulty in maintaining a clear distinction between two Sinitic languages.

Another recurring issue concerns incorrect or distorted semantic interpretation in LLM-generated translations. For example, the Irish sentence *Rinne an buachaill é féin a fholcadh san abhainn* ('The boy bathed in the river') was translated into Cantonese as 那個男孩把自己淹死在河裡 ('The boy drowned himself in the river'). This output is not only in Mandarin but also introduces an incorrect causative 'drown' meaning instead of 'bathe'. Similarly, in translating *Tá an t-ádh ar cibé a cheannaigh an tarracóir seo* ('Whoever bought this tractor is a lucky person'), the model produced 呢個買家買咗嘅嘢都有好運 ('Whatever this buyer bought is lucky'), suggesting a misanalysis of the relative clause structure. In another case, *Chuir an feirmeoir iallach ar a mhac fágáil* ('The farmer made his son leave') was rendered as 農夫把小豬擺喺佢個仔身上 ('The farmer put a pig on his boy'), reflecting a failure to interpret the expression *Chuir ... iallach ar* ('to impose a constraint on/to force').

Invented words and incorrect translation of words were a common feature of the Irish translations. However, the level of inconsistency was surprising, and was clearly context-dependent. Of the 50 sentences in this corpus, 25 involve the word *tractor*. In 6 cases, the word *tarracóir* was correctly translated as 'tractor', but in 19 cases, it was variously translated as 'tarpaulin', 'driver', 'robber', 'burglar', 'rabbit', 'dealer', 'butcher', 'tarrock (a young gull)', 'translator', 'thief', and 'printer'. These predictions appear to be closely related to the context, e.g., 'printer' rather than 'tractor' in 'There is a problem with the ...' Other cases appear to be based on similarity to an English word, e.g., 'tarrock' for *tarracóir* , or an Irish word, e.g., *rósta* 'roasted' for *reoite* 'frozen'.

In the Cantonese-to-Irish translations, we observed a number of morphologically well-formed Irish neologisms. For example, the invented word *tráchtóir* was the most common translation for 'tractor', as well as *tróchar, trachtar, traicéad*, and *tráctor*. This phenomenon was previously reported for Irish by Castilho et al. (2025) and for Faroese by Scalvini et al. (2025).

3.5 Discussion: LLM-generated f-structures

In general, Irish sentences that mirror the predicate–argument structure of their English counterparts fare best, e.g., an active sentence with the ditransitive verb 'give' *Thug an feirmeoir seantarracóir dá chomharsa* ('The farmer gave an old tractor to his neighbour') is analyzed correctly, despite incorrect translations for *seantarracóir* as 'senior farmer' and *chomharsa* as 'companion'. However, many Irish sentences employ a construction different to English, e.g., *Lig an feirmeoir cnead as* 'The farmer let a groan out of him' instead of 'The farmer groaned'. This was plausibly but incorrectly guessed as 'The farmer let the herd out'.

Although the LLM did not always produce well-formed f-structures, it occasionally generated plausible alternative analyses that may be useful to grammar engineers. For example, in the Cantonese ParGram treebank, the prepositional phrase in (6) is analyzed as an ADJUNCT of 覺得 'feel', whereas the LLM treated it as an oblique argument. While this analysis is not adopted in the ParGram grammar, where 覺得 is consistently treated as a two-place predicate, it nonetheless suggests a possible alternative, namely allowing alternation between two-place and three-place structures.

(6) 佢哋 為 個 女 覺得 自豪
3PL for/about CL daughter feel proud
'They feel proud about their daughter.'

We observed that the LLM sometimes employed non-standard labels, such as REL for relative clauses instead of the LFG-preferred ADJUNCT. Although such labels fall outside the standard LFG feature inventory, they still suggest that the underlying structure is at least partially recognizable. LFG distinguishes between functions such as COMP, XCOMP, and PREDLINK (Dalrymple et al., 2019): PREDLINK marks predicative complements, COMP denotes closed complements (e.g., English *that*-clauses), and XCOMP denotes open complements typically found in control constructions (e.g., infinitival complements of *try*). While the LLM produced COMP and XCOMP, it sometimes misapplied these distinctions. In particular, relative clauses were occasionally analyzed as COMP, leading to *Bad* ratings and reflecting a conflation of modification with complementation—two fundamentally distinct notions in linguistic theory. The model also introduced thematic role labels such as AGENT into the f-structure, indicating further confusion between grammatical functions and thematic roles.

We observed potential instances of gender bias in the LLM-generated f-structures. For example, Cantonese nouns such as 農夫 'farmer' and 司機 'driver' were assigned masculine gender values, even though Cantonese does not mark grammatical gender. No contextual information in ParGramBank licenses such interpretations. For both Cantonese and Irish, the model appeared to assign gender features based on stereotypical world knowledge rather than linguistic evidence in the sentence. However, since the prompts did not explicitly specify the ParGram feature inventory, further investigation is needed to determine whether these assignments reflect genuine gender bias or alternative feature conventions adopted by the model.

4 LLM comparisons and ParGram treebanks

While this study focuses on a single multilingual LLM, it is important to situate the findings in relation to models with varying exposure to Cantonese and Irish. Some recent systems, e.g., EuroLLM (Ramos et al., 2026), NLLB (Team, 2024), and Irish-focused models such as UCCIX (Tran et al., 2024) and Qomhrá (McInerney et al., 2026), report greater incorporation of Irish data, while broadly multilingual models include varying amounts of Cantonese (Jiang et al., 2025). In contrast, large proprietary models (e.g., Claude (Enis and Hopkins, 2024)) are trained on substantially larger and more diverse datasets, which may influence performance on low-resource language pairs. Although a systematic comparison is beyond the scope of this paper, the usefulness of LLMs for grammar engineering likely depends on the quantity and quality of language-specific data, as well as task-specific adaptation. In this context, linguistically precise resources such as ParGram treebanks remain crucial: they provide gold-standard analyses for evaluation and benchmarks for assessing whether LLMs capture the formal generalizations required for grammar engineering.

Future work could explore cross-lingual transfer approaches that leverage high-resource languages such as English and Mandarin to support grammar engineering for lower-resource language pairs. Future work may also test models with more targeted language-specific training and fine-tuning, as well as varied prompting strategies and the use of LLMs in different collaborative grammar engineering settings.

5 Conclusions

Grammar engineering requires expertise in both linguistic formalism and computational implementation. This paper has presented the development of Cantonese and Irish ParGram grammars and treebanks, which maintain parallelism at an abstract functional level, alongside an evaluation of a multilingual LLM on translation and LFG-style f-structure generation. While the LLM produced some structurally meaningful outputs, overall performance was limited, particularly in shared f-structure tasks requiring cross-linguistic abstraction. Nonetheless, its outputs may still offer some reference value: even partially correct f-structures can capture aspects of predicate–argument relations and suggest alternative analyses. However, such outputs require careful manual verification and cannot replace expert-driven grammar development.

Acknowledgments

The first author would like to acknowledge the support of a Research Project Grant from the Leverhulme Trust for the project *Modelling Coreference Resolution across Syntax, Semantics, and Discourse* (RPG-2025-064). Both authors are grateful to Mary Dalrymple for her feedback during

the preparation of this manuscript. We also thank members of the ParGram f-structure comparison meetings for their continuous feedback and support since 2023. We are grateful to our language consultants, Gearóid Ó Donnchadha and Lun Kin Lo, for their assistance with the translation rating tasks. Finally, we thank the three anonymous reviewers for their helpful comments and suggestions, and the BriGap program committee for their consideration of this work.

References

Gaper Begu, Maksymilian Dbkowski, and Ryan Rhodes. 2025. Large Linguistic Models: Investigating LLMs' Metalinguistic Abilities. *IEEE Transactions on Artificial Intelligence*, 6(12):3453–3467.

Emily Bender. 2008. Grammar engineering for linguistic hypothesis testing. In *Proceedings of the Texas Linguistics Society X Conference: Computational Linguistics for Less-Studied Languages*, pages 16–36. CSLI Publications, Stanford.

Emily M. Bender, Dan Flickinger, and Stephan Oepen. 2011. Grammar engineering and linguistic hypothesis testing: Computational support for complexity in syntactic analysis. *Language from a cognitive perspective: grammar, usage and processing*, pages 5–29.

Ziqian Bi, Keyu Chen, Chiung-Yi Tseng, Danyang Zhang, Tianyang Wang, Hongying Luo, Lu Chen, Junming Huang, Jibin Guan, Junfeng Hao, Xinyuan Song, and Junhao Song. 2025. Is gpt-oss good? a comprehensive evaluation of openai's latest open source models. *Preprint*, arXiv:2508.12461.

Joan Bresnan, Ash Asudeh, Ida Toivonen, and Stephen Wechsler. 2016. *Lexical-Functional Syntax*. John Wiley & Sons.

Miriam Butt, Tracy Holloway King, María-Eugenia Niño, and Frédérique Segond. 1999. *A grammar writer's cookbook*. Number no. 95 in CSLI Lecture Notes. CSLI Publications, Stanford, Calif.

Yong Cao, Li Zhou, Seolhwa Lee, Laura Cabello, Min Chen, and Daniel Hershcovich. 2023. Assessing Cross-Cultural Alignment between ChatGPT and Human Societies: An Empirical Study. *arXiv preprint*. Version Number: 2.

Sheila Castilho, Zoe Fitzsimmons, Claire Holton, and Aoife Mc Donagh. 2025. Synthetic fluency: Hallucinations, confabulations, and the creation of IrishWords in LLM-generated translations. In *Proceedings of Machine Translation Summit XX: Volume 1*, pages 287–299, Geneva, Switzerland. European Association for Machine Translation.

Dick Crouch, Mary Dalrymple, Ronald M Kaplan, Tracy Holloway King, John Maxwell, and Paula Newman. 2011. *XLE documentation*. Palo Alto Research Centre.

Mary Dalrymple, John J. Lowe, and Louise Mycock. 2019. *The Oxford reference guide to Lexical Functional Grammar*. Oxford University Press.

Denys Duchier and Yannick Parmentier. 2015. High-level methodologies for grammar engineering, introduction to the special issue. *Journal of Language Modelling*, 3(1).

Maxim Enis and Mark Hopkins. 2024. From llm to nmt: Advancing low-resource machine translation with claude. *Preprint*, arXiv:2404.13813.

Martin Forst and Tracy Holloway King. 2023. Computational implementations and applications. In Mary Dalrymple, editor, *The handbook of Lexical Functional Grammar*. Language Science Press, Berlin.

Annette Hautli-Janisz, Erisa Bytyqia, Pranshu Guptaa, and Miriam Butt. 2025. Revisiting the Light Verb Jungle: How Good is GPT Hacking Away? In Aikaterini-Lida Kalouli, Tracy Holloway King, and Annie Zaenen, editors, *Semantics at the Crossroads. From Theoretical Explorations to Implementations*, pages 237–263. PubliKon, Konstanz.

Jiyue Jiang, Pengan Chen, Liheng Chen, Sheng Wang, Qinghang Bao, Lingpeng Kong, Yu Li, and Chuan Wu. 2025. How well do LLMs handle Cantonese? benchmarking Cantonese capabilities of large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4464–4505, Albuquerque, New Mexico. Association for Computational Linguistics.

Ronald M Kaplan and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In *The Mental Representation of Grammatical Relations*, pages 173–281. MIT Press, Cambridge, MA.

Mary Kennedy. 2025. Evidence of Generative Syntax in LLMs. In *Proceedings of the 29th Conference on Computational Natural Language Learning*, pages 377–396, Vienna, Austria. Association for Computational Linguistics.

Tracy Holloway King. 2004. Starting a ParGram Grammar.

Sabine Lehmann, Stephan Oepen, Klaus Netter, and Judith Klein. 1997. TSNLP—test suites for natural language processing. *Linguistic databases*, pages 13–36.

Stephen Matthews and Virginia Yip. 2013. *Cantonese: A comprehensive grammar*. Routledge.

John Philip McCrae and Adrian Doyle. 2019. Adapting term recognition to an under-resourced language: The case of Irish. In *Proceedings of the Celtic Language Technology Workshop*, pages 48–57.

Joseph McInerney, Khanh-Tung Tran, Liam Lonergan, Ailbhe Ní Chasaide, Neasa Ní Chiaráin, and Barry Devereux. 2026. Qomhra: A bilingual irish and english large language model. *Preprint*, arXiv:2510.17652.

Elliot Murphy, Evelina Leivada, Vittoria Dentella, Raquel Montero, Fritz Günther, and Gary Marcus. 2025. Fundamental principles of linguistic structure are not represented by ChatGPT. *Biolinguistics*, 19:e19021.

OpenAI, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, and 107 others. 2025. gpt-oss-120b & gpt-oss-20b Model Card. *arXiv preprint*. ArXiv:2508.10925 [cs].

Miguel Moura Ramos, Duarte M. Alves, Hippolyte Gisserot-Boukhlef, João Alves, Pedro Henrique Martins, Patrick Fernandes, José Pombal, Nuno M. Guerreiro, Ricardo Rei, Nicolas Boizard, Amin Farajian, Mateusz Klimaszewski, José G. C. de Souza, Barry Haddow, François Yvon, Pierre Colombo, Alexandra Birch, and André F. T. Martins. 2026. Eurollm-22b: Technical report. *Preprint*, arXiv:2602.05879.

Victoria Rosén. 2023. LFG treebanks. In Mary Dalrymple, editor, *The Handbook of Lexical Functional Grammar*. Language Science Press.

Victoria Rosén, Koenraad De Smedt, Paul Meurer, and Helge Dyvik. 2012. An open infrastructure for advanced treebanking. In Jan Haji, Koenraad De Smedt, Marko Tadi, and António Branco, editors, *META-RESEARCH Workshop on Advanced Treebanking at LREC2012*, pages 22–29. LREC2012 Workshop.

Barbara Scalvini, Iben Nyholm Debess, Annika Simonsen, and Hafsteinn Einarsson. 2025. Rethinking low-resource MT: the surprising effectiveness of fine-tuned multilingual models in the LLM age. In *Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025)*, pages 609–621, Tallinn, Estonia. University of Tartu Library.

Piyapath T. Spencer and Nanthipat Kongborrirak. 2025. Can LLMs Help Create Grammar?: Automating Grammar Creation for Endangered Languages with In-Context Learning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10214–10227, Abu Dhabi, UAE. Association for Computational Linguistics.

Sebastian Sulger, Miriam Butt, Tracy Holloway King, Paul Meurer, Tibor Laczkó, György Rákosi, Cheikh Bamba Dione, Helge Dyvik, Victoria Rosén, Koenraad De Smedt, Agnieszka Patejuk, Özlem Çetinolu, Wayan Arka, and Meladel Mistica. 2013. ParGramBank: The ParGram Parallel Treebank. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 550–560. Association for Computational Linguistics.

Maggie Tallerman. 2020. *Understanding syntax*, 5th, revised and updated edition edition. Understanding language series. Routledge, London New York, NY.

NLLB Team. 2024. Scaling neural machine translation to 200 languages. *Nature*, 630:841–846.

Khanh-Tung Tran, Barry O'Sullivan, and Hoang D. Nguyen. 2024. Uccix: Irish-excellence large language model. *Preprint*, arXiv:2405.13010.

Rong Xiang, Emmanuele Chersoni, Yixia Li, Jing Li, Chu-Ren Huang, Yushan Pan, and Yushi Li. 2024. Cantonese natural language processing in the transformers era: a survey and current challenges. *Language Resources and Evaluation*.

Olga Zamaraeva, Chris Curtis, Guy Emerson, Antske Fokkens, Michael Goodman, Kristen Howell, T.J. Trimble, and Emily M. Bender. 2022. 20 years of the Grammar Matrix: cross-linguistic hypothesis testing of increasingly complex interactions. *Journal of Language Modelling*, 10(1).

A ParGramBank Sentences

English ParGram Sentences (1–50):
These sentences are openly accessible at: https://clarino.uib.no/iness/home > Treebanks > English and ParGram > View and search the selected treebanks.

1. The driver starts the tractor.
2. The tractor is red.
3. What did the farmer see?
4. Did the farmer sell his tractor?
5. Push the button.
6. Don't push the button.
7. The farmer gave his neighbor an old tractor.
8. The farmer cut the tree down.
9. The farmer groaned.
10. My neighbor was given an old tractor by the farmer.
11. The tree was cut down yesterday.
12. The tree had been cut down.

13. The tractor starts with a shudder.
14. The tractor appeared.
15. The boy knows the tractor is red.
16. The child thinks he started the tractor.
17. The farmer knows who started the tractor.
18. The child wondered whether the button had been pushed.
19. The tractor that the farmer bought is red.
20. The man who bought the tractor left.
21. The store the farmer bought the tractor from closed.
22. Whoever bought this tractor is a lucky person.
23. The farmer made his son clean the tractor.
24. The farmer made his son leave.
25. The farmer made her son buy the tractor.
26. The farmer let her son buy the tractor.
27. The farmer bought his son a tractor.
28. The woman bought the tractor for her husband.
29. The lovers danced until dawn.
30. The boy bathed in the river.
31. The teacher read to himself aloud.
32. The brothers bought the tractor for each other.
33. My sister is a great teacher.
34. The child is in the house.
35. The children are happy.
36. It is raining.
37. There is a problem with the tractor.
38. The book cover depicted a tractor.
39. Let's get ice-cream.
40. The boy swept up the broken wine bottle.
41. The red wine bottle broke.
42. There are great green globs of greasy grimy gopher guts.
43. They are proud of their daughter.
44. My tractor is faster than your sports car.
45. My tractor is the fastest vehicle in the county.
46. The barking dog woke the neighbours.
47. The tea drinking woman admired her new purchase from eBay.
48. The farmer wants to buy a tractor.
49. The farmer's daughter promised to repair the tractor.
50. The farmer persuaded his wife to buy a new tractor.

Cantonese ParGram Sentences (1–50):
The following sentences are the idiomatic Cantonese translation of the respective ParGram sentences.

1. 個司機啓動拖拉機
2. 架拖拉機係紅色嘅
3. 個農夫睇到乜嘢
4. 個農夫賣咗佢嘅拖拉機呀
5. 撳個制
6. 唔好撳個制
7. 個農夫送畀佢嘅隔離屋一架舊拖拉機
8. 個農夫將棵樹斬落嚟
9. 個農夫嘆氣
10. 個農夫送咗一架舊拖拉機畀我隔離屋
11. 棵樹琴日俾人斬咗落嚟
12. 棵樹俾人斬咗落嚟
13. 個拖拉機顫咗一陣之後啓動咗
14. 架拖拉機出現咗
15. 個男仔知道架拖拉機係紅色嘅
16. 個細路仔以為佢啓動咗架拖拉機
17. 個農夫知道邊個啓動咗架拖拉機

18. 個細路仔想知個制係咪俾人撳咗落去
19. 個農夫買嘅拖拉機係紅色嘅
20. 買嗰架拖拉機嘅男人走咗
21. 個農夫買拖拉機嘅舖頭收咗檔
22. 買咗呢架拖拉機嘅人好幸運
23. 個農夫叫佢個仔清理拖拉機
24. 個農夫叫佢個仔走
25. 個農夫叫佢個仔買咗一架拖拉機
26. 個農夫俾佢個仔買咗一架拖拉機
27. 個農夫幫佢個仔買咗一架拖拉機
28. 個女人買咗呢架拖拉機畀佢老公
29. 呢對情侶一直跳舞到天光
30. 個男仔喺條河度沖涼
31. 個老師大聲咁讀畀自己聽
32. 兄弟互相買咗架拖拉機
33. 我家姐係一位好老師
34. 嗰個細路喺屋入面
35. 啲細路好開心
36. 而家落緊雨
37. 架拖拉機有問題
38. 書嘅封面畫咗一架拖拉機
39. 我哋去食雪糕啦
40. 個男仔將酒樽嘅碎片掃起嚟
41. 個紅酒樽爆咗
42. 有一大團油淋淋又污糟嘅地鼠內臟
43. 佢哋為個女覺得自豪
44. 我架拖拉機比你架跑車快
45. 我架拖拉機係成個縣最快嘅交通工具
46. 隻吠緊嘅狗嘈醒咗啲鄰居
47. 個飲緊茶嘅女人欣賞佢喺 eBay 買返嚟嘅新嘢
48. 個農夫想買一架拖拉機
49. 個農夫嘅女應承修理架拖拉機
50. 個農夫勸佢老婆買一架新拖拉機

Irish ParGram Sentences (1–50):
The following sentences are the idiomatic Irish translation of the respective ParGram sentences.

1. Tosaíonn an tiománaí an tarracóir.
2. Tá dath dearg ar an tarracóir.
3. Céard a chonaic an feirmeoir?
4. Ar dhíol an feirmeoir a tharracóir?
5. Brúigh an cnaipe
6. Ná brúigh an cnaipe.
7. Thug an feirmeoir seantarracóir dá chomharsa.
8. Ghearr an feirmeoir an crann anuas.
9. Lig an feirmeoir cnead as
10. Bhí seantarracóir tugtha do mo chomharsa ag an bhfeirmeoir.
11. Gearradh an crann anuas inné.
12. Bhí an crann gearrtha anuas.
13. Tosaíonn an tarracóir le creathán.
14. Nocht an tarracóir
15. Tá a fhios ag an mbuachaill go bhfuil dath dearg ar an tarracóir.
16. Ceapann an páiste gur chuir sé an tarracóir ar siúl
17. Tá a fhios ag an bhfeirmeoir cé a ă chuir an tarracóir ar siúl.
18. Bhí an páiste ag déanamh iontais de ar bhrúdh an cnaipe nó nár bhrúdh.
19. Tá dath dearg ar an tarracóir a cheannaigh an feirmeoir.
20. D'fhág an fear a cheannaigh an tarracóir.
21. Dúnadh an siopa ónar cheannaigh an feirmeoir an tarracóir.
22. Tá an t-ádh ar cibé a cheannaigh an tarracóir seo.

23. Chuir an feirmeoir iachall ar a mhac an tarracóir a ghlanadh.
24. Chuir an feirmeoir iachall ar a mhac fágáil.
25. Chuir an feirmeoir iachall ar a mac an tarracóir a cheannach.
26. Lig an feirmeoir dá mac an tarracóir a cheannach
27. Cheannaigh an feirmeoir tarracóir dá mhac
28. Cheannaigh an bhean an tarracóir dá fear céile
29. Bhí na leannán ag damhsa go breacadh an lae
30. Rinne an buachaill é féin a fholcadh san abhainn
31. Léigh an múinteoir dó féin os ard
32. Cheannaigh na dearthóireacha an tarracóir dá chéile
33. Is múinteoir iontach í mo dheirfiúr
34. Tá an páiste sa teach
35. Tá na páistí sásta
36. Tá sé ag cur báistí
37. Tá fadhb leis an tarracóir
38. Léirigh clúdach an leabhair tarracóir
39. Faighimis uachtar reoite
40. Scuab an buachaillsuasan buidéalfíona briste
41. Bhris an buidéal fíona dheirg
42. Tá blobaí móra glasa de phutóga gófair gréisceacha brocacha ann
43. Tá siad bródúil as a n-iníon
44. Támo tharracóirníostapúla ná do charrspóirt
45. Is é mo tharracóir an fheiticil is tapúla sa chontae
46. Dhúisigh tafann an mhadra na comharsana
47. D'fhéach bean ólta an tae le sásamh ar a ceannachán úr ó eBay
48. Ba mhaith leis an bhfeirmeoir tarracóir a cheannach.
49. Gheall iníon an fheirmeora go ndeiseodh sí an tarracóir.
50. Chuir an feirmeoir ina luí ar a bhean chéile tarracóir nua a cheannach.

B Rating Scheme for F-structure Evaluation

- Excellent (Fully correct)
  - All grammatical functions (GFs) are correctly specified (e.g. SUBJ, OBJ, OBL, COMP, XCOMP, ADJUNCT)
  - Subcategorization frame <...> is correct
  - All relevant non-GF features (e.g., TENSE, ASPECT, NUMBER, GENDER, CASE, PERSON) are correct
- Good (Structurally correct, some incorrect features)
  - All GFs are correct and consistent with the predicate argument structure
  - Subcategorization frame is correct
  - Only issues in non-GF features
- Fair (Partially correct analysis)
  - The core construction is recognizable, but there is at least one GF error, such as a missing argument or incorrect GF assignment (e.g. OBJ vs OBL) or a mismatch with the subcategorization frame
  - May include errors in non-GF features
- Bad (Not recognizable construction)
  - The core construction is not recognizable due to major errors in GFs
  - The f-structure is incompatible with the intended predicate-argument structure; e.g., relative clause or adjunct analyzed as complement

A Formal Model of Lexical Negation in Discrete Communication

Mikołaj Piotr Golecki
Université Paris Cité, CNRS, LLF
mikolaj.golecki@etu.u-paris.fr

Timothée Bernard
Université Paris Cité, CNRS, LLF
timothee-bernard@u-paris.fr

Abstract

Natural languages distinguish between objects satisfying a predicate and those satisfying its complement, often using a simple lexical negation. In emergent communication, however, a system may separate positive and negative meanings without developing a single negation marker: polarity may be tied to the thing being negated, distributed across multiple symbols, or reflected only in accidental correlations. We propose an information-theoretic account of negation applicable to discrete communication systems. We first study these metrics on toy languages, showing how they can be used to detect various patterns and how these are indeed related to negation. We then apply them to languages emerging in a signalling game with set-complement relations, under pressures known to favour compositionality. The results suggest that these pressures can produce high-scoring polarity-sensitive features, but not necessarily a compositional encoding of negation. More generally, we highlight both the usefulness and the limits of targeted semantic diagnostics for analysing structure in emergent languages.

1 Introduction

In natural language, the fact that the meanings of many complex expressions seem to be computable using only the meanings of their immediate subparts and the way they are combined is known as *compositionality* (Frege, 1892; Montague et al., 1970; Partee, 1995; Fodor and Lepore, 2002). Recent work shows that compositionality is not uniformly realised across language use, and that complex expressions in natural language retain a high degree of arbitrariness (Baroni, 2019; Dankers et al., 2022; Sathe et al., 2024). Limited forms of composition are also attested in some animal communication systems (Crockford and Boesch, 2005; Girard-Buttoz et al., 2022; Arnon et al., 2025).

What is distinctive about human language is therefore not simply the presence of regular combination, but its scale (Bortolato et al., 2023), systematicity (Pinker, 2003; Hurford, 2003), and cultural transmission (Kirby et al., 2008; Arnon and Kirby, 2024). Compositionality may be crucial in this respect, since finite, memorable components and reusable rules make it possible to express an infinite range of meanings.

In recent years, substantial work has investigated the emergence of language-like communication systems, including the conditions under which structured and compositional languages arise both in human participants (Kirby et al., 2008, 2014) and computational settings (a.o. Lazaridou et al., 2018; Dessì et al., 2019; Chaabouni et al., 2020; Baroni, 2019; Tucker et al., 2022; Bernard and Mickus, 2023; Akkerman et al., 2024). So far, this work has mostly focused on the emergence of *content words*: labels for objects, properties, spatial relations, commands for movement (e.g. *big blue bug*, *big red cube (to the) top left*, *go right*). Far fewer studies have investigated the emergence of functional or logical elements (Steinert-Threlkeld, 2020; Todd et al., 2020; Korbak et al., 2020), such as quantifiers, determiners, or negation. This is an important gap because content words are commonly interpreted as sets, and their composition via set intersection (e.g. $[\![\text{blue bug}]\!] = [\![\text{blue}]\!] \cap [\![\text{bug}]\!]$) or, equivalently, logical conjunction (e.g. $[\![\text{blue bug}]\!](x) = [\![\text{blue}]\!](x) \wedge [\![\text{bug}]\!](x)$) – the resulting expressions are said to be *trivially compositional* (Schlenker et al., 2016; Zuberbühler, 2018; Steinert-Threlkeld, 2020). By contrast, operators such as negation and quantification introduce non-intersective composition: the meaning of "not blue", for instance, cannot be computed by intersecting the meaning of "blue" with any set. In this article, we focus on negation as a minimal instance of non-intersective operator.

Note however that a language that can express some predicates (or sets) and their negation (or complement) might not encode negation composi-

Third Workshop on the Bridges and Gaps between Formal and Computational Linguistics, pages 135–147

tionally, or not with a single dedicated lexical item or syntactic construction. For instance, negation may be conflated with some predicates (cf. *present* and its negation *absent*) or distributed across multiple lexical items (French *ne ... pas* negation). Distinguishing between these possibilities is necessary for understanding emergent languages, comparing how artificial and natural languages encode negation, characterising typological variation in negation, and analysing how models represent semantic structure.

In this paper, we first propose information-theoretic metrics – *negation strength*, *homogeneity*, *value entanglement*, and *value conservation* – for detecting some forms of negation encoded in discrete communication systems, where signals are finite sequences of symbols. These metrics distinguish constructions that encode polarity compositionally from those that conflate polarity with predicate content, arise from distributed items, or reflect incidental correlations with polarity. We then apply these metrics on controlled toy languages to illustrate which kinds of negative constructions they can be used to detect. Finally, we present a signalling game that crucially revolves around set complementation and apply our metrics on languages obtained in an exploratory setting to study the emergence of negation. The associated experiment serves to illustrate the metrics for emergent language and provides preliminary validation; however, the results remain inconclusive regarding the types of pressure that reliably favour the development of a compositional encoding of negation. Code is available at https://github.com/TimotheeMickus/LEMURIA/.

2 A Formal Model of Negation in Discrete Communication

To express the negation of a predicate, languages sometimes use a different, arbitrary name (ex: *leave* for the negation of *stay*), a prefix (ex: *im* in *impossible*), a preposition (ex: *without*), a syntactic rule involving a dedicated lexical item (ex: *not*), and we can think of many other attested and non-attested solutions (such a simply changing the position of the predicate in the utterance, repeating the name of the predicate, or spelling it backward). In this paper, we focus on forms of negation expressed via one or more elements, be them words, parts of words, or any other sequences of symbols. For simplicity, we call these forms of negation *lexical negations* (in slight contrast with the standard terminology that takes the use of *not* in English to be a form of syntactic negation) to distinguish them from, among other constructions, the use of complex syntactic rules (e.g. ones involving movement or repetition).

In this section we first characterise lexical negation in a discrete communication channel through multiple information-theoretic measures. We then use these measures to define metrics aimed at quantifying negation; optimising said metrics over a set of constructions is a manner of detecting negation. Finally, we discuss examples and cases where the proposed metrics fail to capture intuitively valid encodings of negation, for instance when negation is expressed through multiple lexical items or when correlations unrelated to negation lead to false positives.

2.1 Desiderata for Detecting Negation

Consider a set U and a language $\mathcal{L}$ such that the elements of $\mathcal{L}$, called *signals*, are interpreted as subsets of U. For example, we could have a set of objects U, a signal $\mathrm{r} \in \mathcal{L}$, and r interpreted as the set $[\![\mathrm{r}]\!] \subseteq U$ of red objects. In such a setting, set complementation (in U) can be interpreted as negation: $U \setminus [\![\mathrm{r}]\!]$ is the set of non-red objects; in what follows, we note this set $\neg[\![\mathrm{r}]\!]$. If $\mathcal{L}$ contains many pairs of signals denoting a set and its complement, then these signals might use a lexical item that one would intuitively describe as a negation. How to detect this lexical item automatically?

Here we assume a set of *values* $\mathcal{V}$ (e.g. colours, such as blue and red) in bijection with the objects in U; in fact, we assume that each value is a singleton of U and vice-versa (e.g. each object is of a single colour, and no two distinct objects are of the same colour). We then define the set of *messages* to be $M = \mathcal{V} \cup \{\neg v \mid v \in \mathcal{V}\}$. For instance, if $\mathcal{V} = \{\{o_r\}, \{o_g\}, \{o_b\}\}$, corresponding intuitively to three colours (red, green, and blue), then $M = \{\{o_r\}, \{o_g\}, \{o_b\}, \{o_r, o_g\}, \{o_r, o_b\}, \{o_g, o_b\}\}$.

We assume that the signals are finite sequences of symbols; there is a finite *alphabet* Σ such that $\mathcal{L} \subset \Sigma^*$. We also assume that each signal expresses a message: $\forall s \in \mathcal{L},\ [\![\mathrm{s}]\!] \in M$, so $[\![\mathrm{s}]\!] \subseteq U$.

We call boolean random variables on signals *features* (e.g. the presence of a specific symbol), and our goal is to automatically detect features that are lexical negations as described above. To formalise this notion, we define V and N, the two random variables on messages such that $\forall v \in \mathcal{V}$,

- $V(v) = V(\neg v) = v$,
- $N(v) = +$ and $N(\neg v) = -$.

V is the *value variable* and N the *polarity variable*.

If we consider a set of colours and English restricted to expressions such as *red*, *not red*, *blue*, *not blue*, we observe that the presence of *not* entirely determines N and gives no indication on V. We note this feature $X_A \equiv \texttt{not}$; $X_A = 1$ if *not* appears in the signal, 0 otherwise. In information-theoretic terms (Shannon (1948); Weaver (1953); see also Cover and Thomas (2006) for an introduction) and noting $I(X;Y)$ for the *mutual information* between random variables X and Y, $I(X_A;N)$ is maximal and $I(X_A;V)$ is null.

We argue, however, that a feature X can qualify as a negation even if $I(X;N)$ is not maximal. That $I(X;N)$ be maximal corresponds to a form of *homogeneity*. Imagine a variant of English containing *not*, *red*, and *not-blue* in its vocabulary, such that *red* and *not not-blue* denote red and blue objects respectively, and that *not red* and *not-blue* denote not red and not blue objects respectively. Here, the presence of *not* gives, by itself, no indication on N, but just like in the first language, X_A fully determines N *given* V. In this language, X_A is a non-homogeneous negation. These considerations lead to the following definitions:

Negation Strength

$$n = \frac{I(X;N \mid V)}{H(N \mid V)} \tag{1}$$

Normalising on (conditional) entropy $H(N \mid V)$ coerces the value into a score in $[0,1]$. Candidate features for negation can be selected by optimising this metric.

While we do not take homogeneity to be a necessary property of negation, it is still an interesting one. We quantify it as follows:

Negation Homogeneity

$$h = 1 - \frac{I(X;V \mid N)}{H(V \mid N)} \tag{2}$$

If for some values, the feature X is used to indicate that $N{=}+$, and is used for other values to indicate that $N{=}-$, then given the value of N, knowing whether $X = 1$ helps determining the value of V; in other words, X carries information about V given N. The quantity above is 1 when X is used homogenously and decreases for features that are non-homogeneous.

Negation strength alone can be used to detect negation in simple cases such as the language:

$$\begin{aligned} [\![\text{red}]\!] &= red, \\ [\![\text{blue}]\!] &= blue, \\ [\![\text{not red}]\!] &= \neg red, \\ [\![\text{not blue}]\!] &= \neg blue \end{aligned} \tag{A}$$

Still considering the presence of *not*, X_A perfectly encodes negation and has a negation strength $n = 1$; similarly with $[\![\text{not not-blue}]\!] = blue$ instead of $[\![\text{blue}]\!]$ and $[\![\text{not-blue}]\!] = \neg blue$ instead of $[\![\text{not blue}]\!]$ in the language. However, we are also interested in *composite* features, of the form $X \equiv X_1 \vee X_2 \vee \cdots X_n$ where each X_i is the presence of a lexical item. Consider the following language:

$$\begin{aligned} [\![\text{red}]\!] &= red, \\ [\![\text{blue}]\!] &= blue, \\ [\![\text{not red}]\!] &= \neg red, \\ [\![\text{no blue}]\!] &= \neg blue \end{aligned} \tag{B}$$

In this language, $X_B \equiv (\texttt{not} \vee \texttt{no})$ has maximal negation strength, in contrast with both `not` alone and `no` alone. Considering composite features is useful when negation is lexicalised not by one but by multiple lexical items. But as explained below, when considering composite features, maximal negation strength does not guarantee that the feature under discussion indicates the presence of a negation in the intuitive sense.

Consider for instance the following language:

$$\begin{aligned} [\![\text{red}]\!] &= red, \\ [\![\text{blue}]\!] &= blue, \\ [\![\text{der}]\!] &= \neg red, \\ [\![\text{eulb}]\!] &= \neg blue. \end{aligned} \tag{C}$$

The feature $X_C \equiv \texttt{der} \vee \texttt{eulb}$ has maximal negation strength, however it does not indicate the existence of what we would call a lexical negation. More generally, if there is a bijection between messages and signals, the feature that is the disjunction of all of the signals used for negative messages will always have maximal negation strength (and some other features do too), even if the language is entirely arbitrary and no lexical negation exists.

To help us detect lexical negations, given some composite feature $X \equiv X_1 \vee X_2 \vee \cdots \vee X_n$, we note T the random variable $(X_1, X_2, \cdots, X_n)$ (whose values are in $\{0,1\}^n$). We now define:

Value Entanglement

$$t = \frac{I(T; V \mid X{=}1)}{H(V \mid X{=}1)} \tag{3}$$

$I(T; V|X{=}1)$ quantifies how much knowing which of the X_is are true tells us about V when $X{=}1$. For X_C in language C, t is maximal, revealing that the disjunct features of X_C together fully predict V. The metrics n and t provide solid ground for detecting negation; when negation strength is high and value entanglement is low, we argue that X is probably a lexical negation marker.

However, zero or low value-entanglement may be too strict a criterion. Indeed, value entanglement can be non-zero for cases of class-specific or *typed* negation[1] (it is the case for X_B in B). For instance, a simplified version of German might use the negation `kein` for nouns and `nicht` for adjectives. Consider the language including

$$\begin{aligned} [\![\text{kein Würfel}]\!] &= \neg cube, \\ [\![\text{kein Kugel}]\!] &= \neg sphere, \\ [\![\text{nicht rot}]\!] &= \neg red, \\ [\![\text{nicht blau}]\!] &= \neg blue, \end{aligned} \tag{D}$$

and their non-negated counterparts. For $X_D \equiv (\texttt{kein} \vee \texttt{nicht})$, $n = 1$ and $t = 0.5 > 0$, while X_D is arguably a high-quality negation marker. We believe that it is a negation marker because while T does carry information about V, this information is entirely redundant with what is found elsewhere in the signals (in the nouns and adjectives).

Let us formalise this idea. Given that we consider features of the form $X \equiv X_1 \vee X_2 \vee \cdots X_n$ where X_i is the random variable indicating the presence of some symbol $x_i \in \Sigma$, we can define R the random variable whose value for a signal s is the sequence of symbols obtained from s by removing all occurrences of any of $x_1, x_2, \cdots, x_n$. For instance, if $X \equiv \texttt{not}$, for the signal *not a white crow*, $R =$ *a white crow*. We can now define:

Value Conservation

$$c = \frac{I(R; V \mid X{=}1)}{H(V \mid X{=}1)} \tag{4}$$

Value conservation quantifies how much the signals, "beyond X" (so to speak), encode V. High n but low c indicate that the feature X under discussion separates the meaning space according to N while the remainder carries little value information. In unrestricted feature spaces, search may exploit such distributions to construct non-minimal or weakly interpretable candidates. We dub this risk *feature fabrication*.

Finally, we distinguish two notions of negation for a candidate feature X, based on the previously defined information-theoretic metrics. First, X is a *strong negation* when $n \approx 1$ and $t \approx 0$, i.e. when it reliably encodes polarity without relying on value-specific lexical items. Second, X is a *weak negation* when $n \approx 1$ and $c \approx 1$, i.e. when it reliably encodes polarity possibly relying on value-specific items but then that encode value-information that is redundant with that carried by the rest of the signals. These two notions are captured by the following definitions:

Strong Negativity

$$F_s = 2\frac{n(1-t)}{n+(1-t)} \tag{5}$$

Weak Negativity

$$F_w = 2\frac{nc}{n+c}. \tag{6}$$

2.2 Further Analysis

We now examine how the proposed metrics behave on other controlled languages.

Negation is typologically diverse in natural language (Dryer, 2013) and its realisation may intuitively reflect the strong/weak distinction proposed above. A relatively accessible example of this is French sentential negation, which in its literary form is often expressed by a preverbal particle *ne* together with a postverbal negative element such as *pas* (generic), *jamais* (*never*), *aucun* (*none*), etc. In simplified French with sentences such as [[x ne est pas un chat]] and [[x ne est aucun chat]] (*x is no/not a cat*), $X_{ne} = \texttt{ne}$ is a perfect, homogenous negation feature while $X_{pas \vee aucun} \equiv (\texttt{pas} \vee \texttt{aucun})$ is a weak, homogenous negation feature, although both appear necessary for expressing negation; the difference is that $X_{pas \vee aucun}$ carries additional information beyond polarity information (see scores in Appendix).

In contrast, by alleviating the constraint on marking nouns or adjectives in D, we obtain the language D' $= \{\text{nicht}, \text{kein}, \emptyset\} \times \{\text{Würfel}, \text{Kugel}, \text{rot}, \text{blau}\}$. The feature $X \equiv (\texttt{nicht} \vee \texttt{kein})$ in this language is a perfect negation marker, since *nicht* and *kein* are perfectly synonymous.

[1] This kind of negation is attested for natural language, see for instance Japanese (Nyberg, 2012, 6.3.1).

Note that in some extreme cases of *incomplete languages*, that is languages that do not contain a signal for all messages, value entanglement and value conservation may be undefined for some features that one might want to call a negation. Considering for instance

$$\begin{aligned} [\![\text{cube}]\!] &= cube, \\ [\![\text{not cube}]\!] &= \neg cube, \\ [\![\text{sphere}]\!] &= sphere, \\ [\![\text{pyramid}]\!] &= pyramid, \end{aligned} \tag{E}$$

feature $X \equiv \texttt{not}$ does look like a negation feature, but t and c are here undefined because $H(V \mid X = 1) = 0$.

Finally, the occurrence of negation might be correlated to other changes. Consider for instance the case of simple past sentences in English, in which the use of negation *not* triggers the apparition of auxiliary *did* and a change in verb form (e.g. from *I went* to *I did not go*). The presence of the past auxiliary or of some verb forms is thus correlated with that of the negative item, which has an impact on the scores of the corresponding features, even though only *not* would intuitively be called a negation. Still, this is not a serious issue, because the distributions of these features correlated with negation are significantly different from that of the actual negation *not*.

3 Application to Emergent Languages

In order to study the emergence of negation, we designed the signalling game (Lewis, 1969) described below. Many turns of the game are played in order to train two artificial agents – the *asker* and the *retriever* – who develop a shared language over the course of their training.

3.1 Game Definition

As above, we start by considering a set of arbitrary values $\mathcal{V}$ (e.g. $\{blue, red, \dots\}$, written $\{v_1, v_2, \cdots, v_{|\mathcal{V}|}\}$ below) in bijection with a set of objects; we see each value as a predicate on objects satisfied by exactly one of them. We then define the set of messages as the set of predicates $\mathcal{V} \cup \{\neg v \mid v \in \mathcal{V}\}$ – where $\neg v$ stands for $\lambda x.\ \neg v(x)$ –, containing each value and its logical negation; these messages/predicates are each assigned a unique index. We call predicates in $\mathcal{V}$ *positive* and predicates in $\{\neg v \mid v \in \mathcal{V}\}$ *negative*. Positive predicates are thus each satisfied by a single object while negative predicates are each satisfied by all objects save one.

At each turn of the game, the index i of a predicate is drawn uniformly at random. The asker observes i and produces a signal, which is a finite sequence of symbols. The retriever receives this signal along with a set of objects and must determine which of them satisfy the predicate corresponding to index i. Note that the objects are entirely symbolic; in our setting, there is one object per value in $\mathcal{V}$ and observing an object for the retriever means observing the associated value. The asker is shown only a categorical index because any representation reflecting the syntactic or semantic structure of the predicates would bias the asker toward a structured language (Słowik et al., 2021; Lazaridou et al., 2018) while our goal is to see whether set interactions as experienced by the retriever can induce compositional behaviour.

Both agents are implemented as neural networks, each built primarily around an LSTM (used as a decoder for the asker and as an encoder for the retriever). The asker includes a *predicate embedding matrix* and when it observes an index i, it accesses its i^{th} vector. Similarly, the retriever includes a *value embedding matrix* and when it observe an object, it accesses the embedding of the value associated with this object. The asker is trained using REINFORCE (Williams, 1992): for each object correctly recognised by the retriever, the asker's parameters are updated to stabilise the association between the observed index and the signal produced (and conversely, for each incorrectly recognised object). The retriever is trained using standard supervised learning to classify each input object as positive or negative given the signal it receives. Through repeated training, the asker can associate a meaning (a predicate/set of objects) with each index, a meaning that it learns to describe with signals that the retriever learns to interpret. In this situation, it is entirely possible that the agents develop a non-compositional language where each index is associated with an arbitrary, non-decomposable signal. Nevertheless, under certain constraints – limited memory, restricted symbol inventory – it might become advantageous for them to develop a compositional language.

The presence of semantically encoded negation in the game creates a communicative pressure (or advantage) for the development of a language that will encode negation systematically; a holistic language is by definition composed of one lexical item

per message that the two agents have to agree on, here twice the number of values, while the development on a lexical negation can reduce the set of lexical item to only one per value plus one for negation.

3.2 Experiments

We conduct two experiments involving pressures known to incentivise compositionality in emergent language due to learning advantages and memory limitations (Spike, 2017, §2.5).

First, we introduce a *retriever reset* period n: every n training epochs the retriever is randomly reinitialised (while the asker is untouched). Kirby et al. (2008, 2014) show that compositionality presents a learning advantage when language is transmitted across generations, more so than within a stable population. In our case, we predict that the retriever regularly "forgetting" the language incentivises the agents to develop a compositional encoding of negation. We call an *epoch* a sequence of 250 *batches*, each batch corresponding to 2048 rounds of the game played in parallel, with two objects per predicate, exactly one satisfying it. We train the agents for 50 epochs across spaces of 2×4, 2×16, and 2×64 predicates (below, we often count in terms of predicate pairs, i.e. in terms of $|V|$) and retriever reset period of 1, 4, 10 and 50 epochs (the latter value is equivalent in practice to no retriever reset). Three randomly initialised runs are done per setting. Signal length is capped at 10 and alphabet size at the number of predicates $+1$ (for an end-of-sequence symbol 0), to allow arbitrary codes.

Then, we repeat the experiment with 2×64 predicates and add an *alphabet penalty* to the asker's training loss, i.e. a term that penalises alphabet usage.[2] The higher its value, the stronger is the pressure for the asker to use as few unique symbols as possible, possibly trading off game performance. A strong enough alphabet penalty would, among other things, encourage the asker to be more consistent, avoiding to use multiple symbols interchangeably (as if they were synonyms). In particular, encoding negation with a shared marker is then more economical than using multiple alternatives.

[2]Within a batch of rounds, for each round, the asker receives a penalty for each symbol occurrence in the signal it produces in this round. All symbols, globally in the batch, lead to the same penalty (set as a hyperparameter λ_{alph}) and this penalty is distributed equally over all occurrences of the same symbol.

We report the value of the metrics defined in Section 2 for the features that maximise them when the agents game performance is the highest (one evaluation is performed after each training epoch), defined as the accuracy of the retriever. We focus on this epoch because the metrics are only meaningful for a language that solves the communication task, and because, this study being exploratory, we do not track the full learning trajectory but aim to demonstrate the application of our metrics on successful protocols. Importantly, we do not examine all possible features (which would be intractable) but implement a bounded expansion search: after all non-composite features are built, growth is iterative, only the top-k (here: 72) features (ranked by negation strength) are kept as parents of the next expansion round, and disjunctions are only built up to 8 disjuncts. These should be interpreted as evidence only *within* this tractable subspace rather than exhaustive optima over all possible features. Appendix A indicates which epochs were analysed for each setup. For reference, Figure 3 there compares the scores discussed here with those at the final epoch: for the 64-pair setup, from which our main observations are drawn, the two are close (F_s differences below 0.06, with no systematic direction), indicating that the selected epoch is not an unrepresentative point in training.

3.3 Results

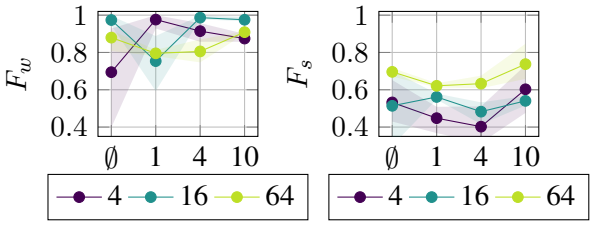


Figure 1: Mean top X F-scores with min–max range over 3 runs, by retriever reset period, with one curve per number of predicate pairs.

Retriever reset. Figure 1 show mean negation F-scores (F_s and F_w) with min-max values over the 3 seeds, depending on the number of predicate pairs and retriever reset period. Recall that a period of 50 effectively amounts to no retriever reset being performed. According to our definition of weak and strong negation, the weak kind dominates overall, though F_s is highest in the 64-pair setup. Within this setup, the 10-epoch reset period yields the highest mean F_s among the four tested intervals,

based on a limited number of seeds. In these runs, the absence of retriever reset (period 50) does not appear to systematically lower the negation score of the top items.

Table 1 provides an example of the realisation of a relatively strong negation feature X ($F_s \approx 0.85$) from this 64-predicate pair setup. This language effectively uses $|\Sigma| = 89$ symbols, and the negation feature is a disjunction over 8 items.

v	**signal for** v	**signal for** $\neg v$
v_1	`114 0`	**`4`** `71 4 15 46 0`
v_2	`114` **`4`** `114 0`	`102 28 96 2 26 0`
v_3	`114 114 14 65 68 43 65 0`	**`126`** `2 88 20 88 0`

Table 1: Selected signal pairs where $X \equiv (1 \vee 126 \vee 3 \vee 36 \vee 4 \vee 41 \vee 83 \vee 85)$ occurs (on the negated, non-negated, and negated side, respectively – see bold symbols). Each line shows the signal produced for each of two predicates; a positive predicate on the left and the corresponding negative predicate on the right.

The size of the disjunction makes the resulting feature difficult to interpret as a compact lexical marker. At the same time, its scores are consistent with a relatively strong negation; n and c are both high (around 0.9), and value entanglement is low (around 0.18) – the symbols on which X relies do express some information beyond negation, but not much. It seems to us that the use of many different symbols here is due to a lack of incentive for the asker to reduce its use of the alphabet at its disposal.

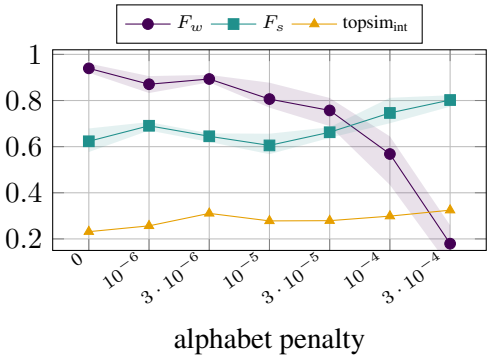


Figure 2: Mean top-1 negation scores with min–max range over 3 runs and mean message-length-normalised Levenshtein topographic similarity, by alphabet penalty, for a 64-predicate-pair setup with retriever reset period=10 over 50 epochs.

Alphabet penalty. We apply an alphabet penalty on the setup with the highest-ranking negation in terms of F_s (64 predicate pairs, retriever reset period = 10) and observe how this affects the languages that then emerge. We report the results for seven non-zero values for the hyperparameter λ_{alph}, up to $3 \cdot 10^{-4}$. Relative to the baseline ($\lambda_{\text{alph}} = 0$, see Table 2), game accuracy remains near-perfect up to $\lambda_{alph} = 10^{-4}$, but drops substantially at $3 \cdot 10^{-4}$; for even larger tested values, the penalty becomes so strong as to lead the asker to produce empty signals (starting and ending with the end-of-sequence symbol 0).[3] This indicates that a moderate pressure toward low alphabet use does not prevent successful communication, while a stronger pressure leads to degenerate strategies.

Across our selected range of λ_{alph} values, F_w decreases overall as the penalty increases (save a non-monotone dip at the lowest non-zero value), with a sharp drop at higher values. By contrast, F_s shows a non-monotone pattern across the range: the mean over 3 runs decreases initially with alphabet penalty, and then starts rising, dominating F_w for the two highest penalty values. This is consistent with high alphabet penalties reducing value entanglement, yielding a stronger negation. Taken together, these trends point to a trade-off. Alphabet penalty constrains vocabulary use, which correlates with a decrease in value entanglement (t, associated with higher F_s) and a decrease in value conservation (c, associated with lower F_w): symbol reuse then appears to promote a more general polarity marking at the cost of compositional separability of predicate-value information. At the highest tested penalty the emergent language approaches degeneracy and the F_s gain there may be a consequence of vocabulary compression and not actual emergence.

Figure 2 also shows topographic similarity (Brighton and Kirby, 2006), the correlation between distances between messages on the one hand and distances between the signals produced to convey the messages on the other hand. Topographic similarity (*topsim*) is computed as Spearman's ρ between pairwise message distances – defined here as the cosine distance between the asker's predicate embeddings – and pairwise signal distances – defined here as the normalised Levenshtein distance on signals – for the language obtained at the end of a training epoch.[4] We include it as a reference

[3] See the previous footnote concerning λ_{alph}. For comparison, the base reward is $\pm\frac{1}{2}$ for each object (in)correctly classified by the retriever.

[4] Because we define message distances as distances between the model's learned predicate embeddings, we call the resulting topographic similarity *intensional*. Instead of these learned embeddings, an *extensional* alternative would consist in using the boolean vectors indicating, for a given list of objects, which ones satisfy the predicates. We do not find this

alpha. penalty	F_w	ΔF_w	F_s	ΔF_s	acc.	Δacc.
baseline	0.909	+0.000	0.741	+0.000	1.000	+0.000
0	0.939	+0.030	0.623	-0.118	0.999	-0.000
10^{-6}	0.871	-0.038	0.691	-0.050	0.999	0.000
$3 \cdot 10^{-6}$	0.893	-0.016	0.645	-0.096	0.999	-0.001
10^{-5}	0.806	-0.103	0.605	-0.136	1.000	0.000
$3 \cdot 10^{-5}$	0.757	-0.152	0.662	-0.079	0.994	-0.006
10^{-4}	0.568	-0.341	0.746	+0.005	0.990	-0.010
$3 \cdot 10^{-4}$	0.179	-0.730	0.802	+0.061	0.842	-0.158

Table 2: Game accuracies and top X mean F-scores for vocabulary penalty values relative to the baseline from the first experiment at properties $= 64$ and retriever reset interval $= 10$. Deltas are alpha-pen. minus baseline.

point because it is common in emergent language work. In the present experiments, topographic similarity shows only limited variation across alphabet-penalty conditions. This is consistent with previous work reporting that topographic similarity fails to detect non-trivial forms of compositionality such as the one involved with negation (Korbak et al., 2020).

v	**signal for v**	**signal for $\neg v$**
v_1	`28 0`	**`99 99`** `0`
v_2	**`99 99`** `0`	`93 0`
v_3	**`78`** `0`	**`99 99`** `0`

Table 3: Selected signal pairs where $X \equiv (78 \vee 99)$ occurs, in a game with $\lambda_{alph} = 3 \cdot 10^{-4}$.

The following examples suggest that in some settings the current feature-selection procedure can recover high-scoring, composite features, whose interpretation as compositional negation markers remains unclear.

The case at $\lambda_{alph} = 3 \cdot 10^{-4}$ illustrates the vocabulary compression effect. The emergent language is not entirely degenerate, but still highly constrained: symbol `99`, involved in the identified feature $X \equiv (78 \vee 99)$, is also the most frequent symbol, accounting for about 45% of all tokens. Table 3 shows that X often separates negative from positive values, but non-homogeneously and in fact non-systematically. In the first row, X is active only on the negative side through `99`; in the second, only on the positive side through `99`; and in the third, both on the positive side through `78` and the negative side through `99`. Thus, X tracks polarity to a certain extent (n is high) without corresponding to a clean lexical negation item.

The language obtained at $\lambda_{alph} = 10^{-4}$ (Table 4) yields much better game accuracy, but also heavily makes use of a few frequent symbols in many signals, including the symbol `117`, involved in the detected composite feature. We note however that the second pair in Table 4 contains the feature on both sides; the other pairs contain it only on one side. The feature's high F_s suggests a strong negation according to our definitions, but its realisation is not uniform across pairs, which makes it difficult to interpret as a lexical marker.

v	**signal for v**	**signal for $\neg v$**
v_1	**`78`** `74 0`	`15 15 0`
v_2	**`117 117`** `4 0`	**`117`** `3 0`
v_3	`29` **`34`** `29 0`	`29 29 0`

Table 4: Selected signal pairs where $X \equiv (117 \vee 34 \vee 36 \vee 78)$ occurs, in a game with $\lambda_{alph} = 10^{-4}$.

4 Discussion

The toy-language analysis supports the intended interpretation of the metrics and shows that they distinguish different negation encodings and degenerate features. In the emergent languages, the main difficulty is to determine whether the highest-scoring candidates correspond to interpretable lexical items. In some successful settings, strong negation is realised by large disjunctions, and qualitative analysis shows that several disjunctions differing by only one or two disjuncts score almost identically. This pattern suggests that our search procedure may recover distributed or non-minimal realizations of negation, rather than the compact markers one might hope to observe. The present exploratory study cannot determine whether this reflects actual distribution in the emergent language or over- or under-expansion in the search procedure.

This suggests that feature selection is also an important issue, beyond metric design. Bounding disjunction size makes the search tractable and may reduce the risk of over-expansion, if it is indeed relevant. Conversely, if a large disjunction is ob-

alternative notion of distance satisfying; for instance, for two distinct values v_1 and v_2, it takes v_1 and v_2 to be very close, and v_1 and $\neg v_1$ to be maximally distant.

tained by gradual expansion of an initially good candidate, and each added item only marginally improves negation strength, the score may be only partially fabricated rather than entirely spurious. Investigating this trade-off more directly appears to be an important direction for future work.

The fact that topographic similarity does not target the encoding of any specific semantic operation (in particular negation) justifies the use of targeted metrics such as ours. Meaning-form correlation may miss local structure tied to a specific semantic operation. At the same time, high negation metric scores do not imply that the language as a whole is compositionally organised. Global and targeted measures may capture different aspects of structure and should be interpreted jointly.

5 Limitations

This study has several limitations. This framework is designed for discrete communication channels with symbol-based features, and does not directly extend to settings where polarity is encoded in continuous representations. The formal setup is limited to atomic unary predicates and their complement/negation, and therefore does not cover richer semantic phenomena such as scope interactions. The empirical study is exploratory and computationally bounded, with few runs and a restricted top-k pruning strategy, rather than exhaustive feature discovery. The current search space is restricted to single symbol-presence features and their disjunctions, and does not consider conjunctions of features, repetitions, and more complex combinations. High-scoring candidates in emergent languages remain difficult to interpret, since disjunctions may reflect genuine distributed encoding, feature fabrication, or both. We do not yet establish principled thresholds for how high t or c must be, nor how strongly they ought to be weighted relative to n, in deciding whether a feature can count as a "good" negation marker. Moreover, candidate selection for feature expansion depends on negation strength, but other metrics, namely t, could play an important role in selection. Many computational pressures associated with the emergence of compositionality have not been considered; in particular, an additional penalty on signal length could complement the alphabet penalty.

References

Daniel Akkerman, Phong Le, and Raquel G. Alhama. 2024. The Emergence of Compositional Languages in Multi-entity Referential Games: from Image to Graph Representations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18713–18723, Miami, Florida, USA. Association for Computational Linguistics.

Inbal Arnon and Simon Kirby. 2024. Cultural evolution creates the statistical structure of language. *Scientific Reports*, 14(1):5255.

Inbal Arnon, Simon Kirby, Jenny A. Allen, Claire Garrigue, Emma L. Carroll, and Ellen C. Garland. 2025. Whale song shows language-like statistical structure. *Science*.

Marco Baroni. 2019. Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 375(1791):20190307.

Timothée Bernard and Timothee Mickus. 2023. So many design choices: Improving and interpreting neural agent communication in signaling games. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8399–8413, Toronto, Canada. Association for Computational Linguistics.

Tatiana Bortolato, Roger Mundry, Roman M. Wittig, Cédric Girard-Buttoz, and Catherine Crockford. 2023. Slow development of vocal sequences through ontogeny in wild chimpanzees (pan troglodytes verus). *Developmental Science*, 26(4):e13350.

Henry Brighton and Simon Kirby. 2006. Understanding linguistic evolution by visualizing the emergence of topographic mappings. *Artificial Life*, 12(2):229–242.

Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. 2020. Compositionality and Generalization In Emergent Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4427–4442, Online. Association for Computational Linguistics.

Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience.

Catherine Crockford and Christophe Boesch. 2005. Call combinations in wild chimpanzees. *Behaviour*, 142(4):397–421.

Verna Dankers, Elia Bruni, and Dieuwke Hupkes. 2022. The Paradox of the Compositionality of Natural Language: A Neural Machine Translation Case Study. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4154–4175, Dublin, Ireland. Association for Computational Linguistics.

Roberto Dessì, Diane Bouchacourt, Davide Crepaldi, and Marco Baroni. 2019. Focus on What's Informative and Ignore What's not: Communication Strategies in a Referential Game. *arXiv preprint*. ArXiv:1911.01892 [cs].

Matthew S. Dryer. 2013. Negative morphemes (v2020.4). In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Zenodo.

Jerry A Fodor and Ernest Lepore. 2002. *The compositionality papers*. Oxford University Press.

Gottlob Frege. 1892. On sense and reference.

Céline Girard-Buttoz, Emiliano Zaccarella, Tania Bortolato, and 1 others. 2022. Chimpanzees produce diverse vocal sequences with ordered and recombinatorial properties. *Communications Biology*, 5:410.

James R. Hurford. 2003. The language mosaic and its evolution. In Morten H. Christiansen and Simon Kirby, editors, *Language Evolution*, pages 38–57. Oxford University Press.

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.

Simon Kirby, Tom Griffiths, and Kenny Smith. 2014. Iterated learning and the emergence of language. *Current Opinion in Neurobiology*, 28:108–114.

Tomasz Korbak, Julian Zubek, and Joanna Rączaszek-Leonardi. 2020. Measuring non-trivial compositionality in emergent communication. *arXiv preprint*. ArXiv:2010.15058 [cs].

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. 2018. Emergence of linguistic communication from referential games with symbolic and pixel input. *Preprint*, arXiv:1804.03984.

David Lewis. 1969. *Convention: A Philosophical Study*. Harvard University Press, Cambridge, MA.

Richard Montague and 1 others. 1970. Universal grammar. *1974*, pages 222–46.

Joacim Nyberg. 2012. Negation in japanese. Bachelor's thesis, Lund University, Lund, Sweden.

Barbara H. Partee. 1995. Lexical semantics and compositionality. In Lila R. Gleitman and Mark Liberman, editors, *An Invitation to Cognitive Science: Language*, pages 311–360. MIT Press, Cambridge, MA.

Steven Pinker. 2003. Comprehension and production in language evolution. In Morten H. Christiansen and Simon Kirby, editors, *Language Evolution*, pages 16–37. Oxford University Press.

Abhijit Sathe, Evelina Fedorenko, and Noga Zaslavsky. 2024. Language use is only sparsely compositional: The case of english adjective-noun phrases in humans and large language models. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46.

Philippe Schlenker, Emmanuel Chemla, Anne M. Schel, James Fuller, Jean-Pierre Gautier, Jeremy Kuhn, Dunja Veselinović, Kate Arnold, Cristiane Cäsar, Sumir Keenan, Alban Lemasson, Karim Ouattara, Robin Ryder, and Klaus Zuberbühler. 2016. Formal monkey linguistics. *Theoretical Linguistics*, 42(1-2):1–90.

Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Agnieszka Słowik, Abhinav Gupta, William L. Hamilton, Mateja Jamnik, Sean B. Holden, and Christopher Pal. 2021. Structural inductive biases in emergent communication. *Preprint*, arXiv:2002.01335.

Matthew John Spike. 2017. *Minimal Requirements for the Cultural Evolution of Language*. Ph.D. thesis, The University of Edinburgh.

Shane Steinert-Threlkeld. 2020. Toward the emergence of nontrivial compositionality. *Philosophy of Science*, 87(5):897–909.

Graham Todd, Shane Steinert-Threlkeld, and Christopher Potts. 2020. Learning compositional negation in populations of roth-erev and neural agents. *Preprint*, arXiv:2012.04107.

Mycal Tucker, Roger Levy, Julie A Shah, and Noga Zaslavsky. 2022. Trading off utility, informativeness, and complexity in emergent communication. In *Advances in Neural Information Processing Systems*, volume 35, pages 22214–22228. Curran Associates, Inc.

Warren Weaver. 1953. Recent contributions to the mathematical theory of communication. *ETC: a review of general semantics*, pages 261–281.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.

Klaus Zuberbühler. 2018. Combinatorial capacities in primates. *Current Opinion in Behavioral Sciences*, 21:161–169.

A Appendix

msg. pairs	retriever reset interval	n	mean epoch	min	max	unique epochs
4	$\emptyset$	3	0.0	0	0	0
4	1	3	0.7	0	1	0, 1
4	4	3	2.0	0	6	0, 6
4	10	3	0.3	0	1	0, 1
16	$\emptyset$	3	17.7	6	40	6, 7, 40
16	1	3	19.0	2	45	2, 10, 45
16	4	3	23.3	3	49	3, 18, 49
16	10	3	13.0	5	26	5, 8, 26
64	$\emptyset$	3	34.0	27	43	27, 32, 43
64	1	3	46.7	44	49	44, 47, 49
64	4	3	33.0	30	35	30, 34, 35
64	10	3	44.0	35	49	35, 48, 49

Table 5: Epochs analyzed for the finite retriever reset interval-only study F-score plots (best accuracy).

voc. penalty	n	mean epoch	min	max	unique epochs
0	3	32.3	19	49	19, 29, 49
10^{-6}	3	35.0	28	49	28, 49
$3 \cdot 10^{-6}$	3	35.3	29	39	29, 38, 39
10^{-5}	3	30.0	16	49	16, 25, 49
$3 \cdot 10^{-5}$	3	41.3	26	49	26, 49
10^{-4}	3	33.3	16	46	16, 38, 46
$3 \cdot 10^{-4}$	3	48.0	46	49	46, 49

Table 6: Epochs analyzed for the voc-penalty study F-score plots (best accuracy). Message pairs=64, retriever reset interval=10.

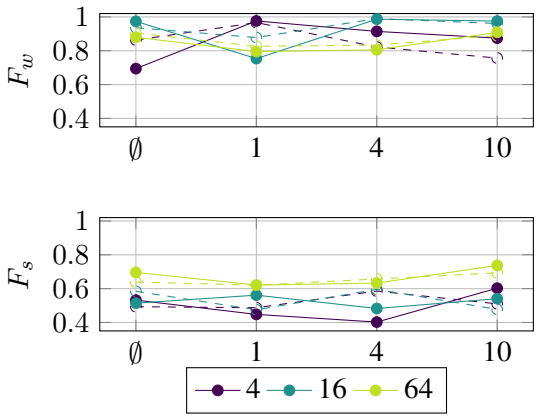


Figure 3: Mean negation F-scores at peak-accuracy epoch (solid, filled) and final epoch (dashed, open), by retriever reset period, with one curve per number of predicate pairs. See Figure 1 for the main result.

Cat sentences				
P0-v0	I am a cat		I am not a cat	
P0-v1	the cat sat on the mat		the cat did not sit on the mat	
P0-v2	the cat is hairy		the cat is not hairy	
P0-v3	curiosity killed the cat		curiosity did not kill the cat	
feat.	n	h	t	c
`not`	1	1	0	1
`did`	0.5	0.75	0	1
`kill`$\vee$`sit`	0.5	0.75	1	1
`cat`	0	1	0	1
French cats				
P0-v0	je suis un chat *I am a cat*		je ne suis pas un chat | je ne suis aucun chat *I am not a cat* | *I am no cat*	
P0-v1	le chat a déjà été sur la branche *the cat has already been on the branch*		le chat ne a jamais été sur la branche *the cat has never been on the branch*	
P0-v2	le chat a été chez le docteur *the cat went to the doctor*		le chat ne a pas été chez le docteur *the cat did not go to the doctor*	
feat.	n	h	t	c
`ne`	1	1	0	1
`pas`	0.41	0.81	0	1
`aucun`$\vee$`jamais`$\vee$`pas`	1	1	0.67	1
`aucun`$\vee$`déjà`$\vee$`pas`	1	0.44	0.67	1
`déjà`	0.3	0.74	$\emptyset$	$\emptyset$
Twisted logic				
P0-v0	I am present | I am not absent		I am not present | I am absent	
P0-v1	I am anxious | I am not calm		I am not anxious | I am calm	
P0-v2	I am here		I am not here	
feat.	n	h	t	c
`absent`$\vee$`calm`$\vee$`not`	0.45	0.94	0.53	0.61
`not`	0.2	0.89	0	1
`absent`$\vee$`calm`	0	0.89	1	0
Untwisted logic				
P0-v0	I am present		I am not present	
P0-v1	I am anxious		I am not anxious	
P0-v2	I am here		I am not here	
P1-v0	I am not absent		I am absent	
P1-v1	I am not calm		I am calm	
feat.	n	h	t	c
`not`	1	0.58	0	1
`I`	0	1	0	1
`anxious`	0	0.69	$\emptyset$	$\emptyset$

Table 7: Natural-language illustration with feature scores.

$\mathcal{L}_1$ **Perfect negation**					**+**	**-**
P0-v0					1	4 1
P0-v1					2	4 2
P0-v2					3	4 3
feat.	n	h	t	c		
4	1	1	0	1		
$\mathcal{L}_2$ **Holistic encoding**					**+**	**-**
P0-v0					1	4
P0-v1					2	5
P0-v2					3	6
feat.	n	h	t	c		
1∨2∨3	1	1	1	0		
4∨5∨6	1	1	1	0		
1∨2∨6	1	0.42	1	0		
1∨2	0.67	0.71	1	0		
$\mathcal{L}_2'$ **Disjoint holistic enc.**					**+**	**-**
P0-v0					1	4 7
P0-v1					2	5 8
P0-v2					3	6 9
feat.	n	h	t	c		
4∨5∨6	1	1	1	1		
4∨5∨9	1	1	1	1		
7∨8∨9	1	1	1	1		
$\mathcal{L}_2''$ **D. h. enc. with variation**					**+**	**-**
P0-v0					1	4 7
P0-v1					2	5 8
P0-v2					3	6 9 ∣ 4 9
feat.	n	h	t	c		
4∨5∨6	1	1	0.67	1		
4∨6∨8	1	1	0.67	1		
2∨4∨6	1	0.44	0.67	1		
1∨5∨9	1	0.44	1	1		
$\mathcal{L}_3$ **Negation variants**					**+**	**-**
P0-v0					1	5 1
P0-v1					2	5 2
P0-v2					3	6 3
P0-v3					4	6 4
feat.	n	h	t	c		
5∨6	1	1	0.5	1		
5	0.5	0.75	0	1		
6	0.5	0.75	0	1		
$\mathcal{L}_4$ **Inhomogenous negation**					**+**	**-**
P0-v0					1	4 1
P0-v1					4 2	2
P0-v2					3	4 3
feat.	n	h	t	c		
4	1	0.42	0	1		
1∨2∨3	0	0.42	$\emptyset$	$\emptyset$		

Table 8: Toy languages with negation-metric scores.

A graph-based analysis of semantic types and coercion in contextualized word embeddings

Long Chen **Deniz Ekin Yavas**
Heinrich Heine University Düsseldorf
{chen.long,deniz.yavas}@hhu.de

Abstract

Semantic type mismatch between a noun and its context is central to coercion phenomena. This paper introduces a graph-based method to examine how lexical and contextual type information is reflected in contextualized word embeddings. We select nouns from ten semantic types, annotate corpus instances for type matching (matching vs. coercion vs. other mismatch vs. unrestricted), and construct graphs using BERT and sense-enhanced BERT embeddings. Two metrics—Neighbor Type Probability (*NTP*) and Neighbor Type Entropy (*NTE*)—are proposed to analyze neighborhood type distributions. Results show that graphs constructed with sense-enhanced BERT embeddings reflect semantic type information better, and matching and mismatch sentences can be distinguished through the proposed metrics.

1 Introduction

Word meaning is composed of various aspects, and semantic type is a fundamental part of it (e.g. Montague, 1970). It is related not only to the conceptual category of the noun but also its usage in the grammar. Ideally, semantic types can be defined by the distribution of words (Asher, 2011). For example, the noun 'pizza' in (1-a) belongs to the semantic type *food* and typically co-occurs with predicates such as 'delicious' and 'eat'.

(1) a. I am eating a delicious pizza.
b. I finished the pizza.

However, nouns are not always used in their prototypical, literal way. The context of an instance may require a semantic type different from that of the noun itself. Coercion (Pustejovsky, 1993) is a typical case of mismatch between the lexical type, i.e. the semantic type of an instance itself and the context type, i.e. the semantic type the context requires or suggests. In a coercion sentence like (1-b), the noun co-occurs with 'finish', which typically selects for an *activity* rather than a *food*.

In this paper, we investigate how the information about semantic types is reflected in the contextualized word embeddings of noun instances, both in typical cases of normal predication as in (1-a) and in non-canonical cases of coercion as in (1-b). For this purpose, we conduct a graph-based analysis of contextualized word embeddings using cosine similarity between the embeddings to connect instances similar to each other.

We construct a dataset with nouns from ten semantic types and annotate their corpus occurrences as different types of sentences. Four possible types of sentences are distinguished, with matching sentences and coercion sentences being the main focus of our study. A sentence is seen as a matching sentence if the lexical type matches the contextual type, and coercion is a typical case of mismatch between the lexical type and the contextual type.

We experiment with different language model embeddings: BERT embeddings (Devlin et al., 2019) and sense-enhanced BERT embeddings (Yavas et al., 2025). The latter is a fine-tuned variant of BERT in order to incorporate WordNet supersense information (Fellbaum, 1998) into the embeddings. WordNet supersenses correspond well to different semantic types. In addition, we also create graphs using masked versions of these embeddings, where the target word is replaced with a [MASK] token. This allows the focus on the information about contextual types by removing lexical information. We propose two metrics (Neighbor Type Probability (*NTP*) and Neighbor Type Entropy (*NTE*)) to quantify the distribution and diversity of semantic types among each instance's neighbors in the graph.

By comparing the neighbor types across the instances within the same lexical type or the same sentence type, we discover that graphs constructed with sense-enhanced BERT embeddings reflect the

Third Workshop on the Bridges and Gaps between Formal and Computational Linguistics, pages 148–159

semantic types better than the ones with BERT embeddings; the lexical types are reliably reflected by the graphs constructed with sense-enhanced embeddings, while contextual types are also partially reflected by graphs constructed with masked embeddings. Instances in different types of sentences display a different pattern in terms of the types of their neighbors. In matching sentences like (1-a), the instances usually share the same lexical type as their neighbors, while in coercion sentences like (1-b) the instances exhibit a higher diversity of types among the neighbors. These findings indicate the effectiveness of our graph-based method.

Our method also has some potential linguistic applications. For example, a hierarchy of semantic types can be induced from the embeddings using the proposed graph-based method and metrics. Similarly, the method and the metrics can be applied to corpus data in order to discover possible coercion instances.

2 Related work

2.1 Coercion

Coercion has been a key challenge to the theory of semantic types, as the lexical type of a noun in a coercion construction conflicts with the expected types from the context. A range of formal frameworks have been proposed to address this challenge, including Generative Lexicon (GL) (Pustejovsky, 1995), Type Compositional Logic (TCL) (Asher, 2015), Modern Type Theory (MTT) (Luo, 2012), frame semantics (Chen et al., 2022), among others.

These approaches can be broadly divided into two groups according to how they model the mechanisms underlying coercion. The first group, which includes GL and MTT, assumes a shift in the semantic type of the coerced instance itself. In GL, for example, a noun instance in coercion undergoes a type shift, as in (2), the type of the noun 'bottle' shifted from *artifact* to *activity*.

(2) I finished the bottle off in two gulps.

The second group, including TCL and frame semantics, assumes a different semantic type not on the coerced noun but on the predication relation, or more generally speaking, the context around the noun. Within frame semantics, for instance, in (2), the noun *bottle* retains its type *artifact*, and it is the predicate that accepts an *artifact* as its object and creates an eventive reading. Despite the richness of these theoretical proposals, computational work that directly supports or operationalizes them remains comparatively scarce. One notable exception is Asher et al. (2016), who applied distributional models to adjective–noun composition—a domain that includes coercive cases—and provided evidence in favor of TCL.

2.2 Pre-Trained Language Models and Semantic Types and Coercion

Few studies focus on whether semantic type knowledge is encoded in the embeddings of pre-trained language models. These studies train classifiers using the frozen embeddings of pre-trained language models and show high performance (Zhao et al., 2020; Yavas et al., 2023).

Several studies focus on coercion interpretation with pre-trained language models based on their word predictions and more specifically focusing on coercion to *event*. Rambelli et al. (2020) investigate covert event retrieval in cases of coercion in English, evaluating several pre-trained language models against human judgments. Their results show that the models fail to substantially outperform simpler distributional models.

Gietz and Beekhuizen (2022) show that coercion interpretation is not necessarily resolved to a single covert event. This is evidenced by low human annotator consensus on the underlying event for naturally-occurring coercion sentences in English. They test different computational models including BERT, co-occurrence counts, prototype vector, and example-based learning models. BERT's performance is tested using masked word prediction and it outperforms other computational models in both high and low consensus cases. Ye et al. (2022) evaluate coercion interpretation in English using masked word prediction, measuring whether the top-1 and top-3 predictions of BERT match the underlying covert event and they report poor performance.

Radaelli et al. (2025a) investigate covert event interpretation in coercion sentences in Norwegian using word prediction across 17 language models. They evaluate whether in models' predictions plausible events are ranked above less plausible ones. Results show that models fail to systematically and consistently rank plausible events higher, and only few outperform a simple corpus frequency baseline. Radaelli et al. (2025b), following Radaelli et al. (2025a), investigate how additional context affects coercion interpretation. While the models generally benefit from additional context, the

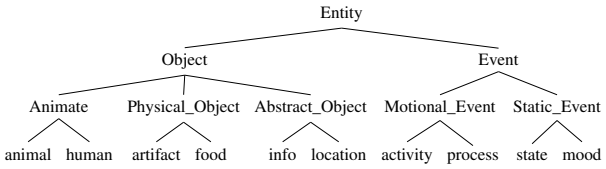


Figure 1: A presumed type hierarchy of the selected semantic types.

performance varies depending on the model. Models that struggle in context-neutral sentences show greater improvements.

These studies investigate coercion focusing on word predictions of the pre-trained language models. To our knowledge, our study is the first to focus on this phenomenon on the representation level. Furthermore, these studies focus on one type of coercion. We examine a broader range of coercion types and investigate how semantic type relations are more generally captured by the contextualized word embeddings of BERT.

3 Data

We select a set $\mathcal{T}$ of ten semantic types for analysis: $\mathcal{T} = \{$*animal*, *artifact*, *activity*, *food*, *human*, *info*, *location*, *mood*, *process*, *state*$\}$. Some of them are ontologically related and can be grouped into higher-level concepts. For example, *human* and *animal* together form the superordinate category of *animate* entities, while *state* and *mood* can be subsumed under *static event*. A conceptual hierarchy of these ten semantic types is shown in Fig. 1. These types are selected because their prototypical instances are relatively easy to distinguish. Moreover, each type is presumably close (in the ontological sense) to some types but not others, which motivates a basic hierarchical structure. This hierarchy is broadly compatible to some existing taxonomies of the relevant types (e.g. WordNet (Fellbaum, 1998)).Currently, we do not include inherently polysemous (dot type) nouns to ensure that the graphs can be organized with instances with one clear lexical type. Given suitable data, this work can be naturally extended to dot type nouns to see how their different lexical types are reflected in the embeddings under different context. We will leave that for future research.

For each type, we select five to ten relatively frequent nouns. Some types have fewer nouns due to the limited number of high-frequency examples.[1]

[1]For example, many nouns associated to the type *info* tend to be more or less related to *artifact*.

For each selected noun, we randomly extract 20 sentences containing the noun from BookCorpus (Zhu et al., 2015). A subset of sentences is manually removed for two reasons. First, some sentences exhibit a similar syntactic and semantic distribution to other existing sentences. Second, certain selected nouns are polysemous and the instance in the extracted sentence corresponds to another sense. After the filtering process, each noun is represented by 10 to 16 sentences. All the selected nouns can be found in the Table 7 and 8 in the appendix, and the full dataset is published in https://github.com/yavasde/Graph-Analysis-Semantic-Types.

Each sentence is annotated with information about the semantic types related to the selected noun independently by two annotators experienced in semantic research on coercion and semantic types.[2] Two notions of semantic types are distinguished: lexical type and contextual type (abbreviated as lt and ct). Lexical type refers to the semantic type of the noun itself, and contextual type refers to the semantic type that the surrounding context requires or suggests. In practice, the contextual type of an instance can be inferred by masking the noun in the sentence. Consider example (3-a), where the lexical type of the noun 'conference' is *activity*. When the noun is masked, as in (3-b), the noun that can felicitously fill the masked position can be 'meeting', 'match', 'argument', all of which are associated to *activity*. Therefore, the contextual type of 'conference' in (3-a) is also *activity*. Such sentences, where $lt = ct$, are annotated as MATCHING. In our dataset, 88% of the sentences (1410 sentences) are labeled as MATCHING, consistent with the general assumption in linguistics that in a sentence, the contextual type matches the lexical type.

(3) a. The **conference** with James Stickley lasted for more than an hour.
(lt: *activity*, ct: *activity*)

b. The [MASK] with James Stickley lasted for more than an hour.
(ct: *activity*)

In addition to the matching cases discussed above, we also encounter instances where $lt \neq ct$, i.e. the semantic type of the noun does not align with the type required or implied from the context. Coercion is a typical case of such a mismatch. In a coercion sentence, the noun combines with a predi-

[2]The agreement between the two annotators is 89.3%.

cate that typically selects for another semantic type. (4-a) is an example of COERCION, where the verb 'roar' conventionally takes an instance of *human* instead of a *location* as its subject. Consequently, in (4-a), the contextual type (ct) of 'stadium' is *human*, while its lexical type (lt) is *location*, resulting in a clear mismatch. These cases are annotated as COERCION. In our dataset, 81 sentences receive this label.

Mismatch between lt and ct can also arise from other mechanisms, such as metaphor (e.g.(4-b)) and metonymy, though such instances are comparatively rare in our sample. We annotate them under the label OTHER_MISMATCH. Sixteen sentences in the dataset are labeled as OTHER_MISMATCH. When these two mismatch types are discussed together without distinction, we refer to them collectively as MISMATCH.

The numbers of MISMATCH sentences are relatively low compared to MATCHING sentences as they are naturally less common in the corpus. However, we do not introduce additional artificial or external examples because we intend to organize the graphs around typical, literal instances, so that the neighborhood composition can be examined without the distraction of too many semantic factors. Despite the low numbers, our COERCION data cover a wider range of coercion phenomena, such as the coercion to *location* and *human*, compared to many of the related resources that focus only on the coercion to *event*.

(4) a. One moment the **stadium** was roaring and the next, everything went completely silent. (lt: *location*, ct: *human*)
b. Just because I'm not a social **butterfly** doesn't mean I'm not smart or capable. (lt: *animal*, ct: *human*)

For all sentences labeled as COERCION or OTHER_MISMATCH, we additionally annotate the contextual type.[3]

Besides the above two cases, there is a third situation that the context is highly general and imposes little selectional restriction on the semantic type of the noun. In such cases we say $ct = \emptyset$. Such contexts are theoretically compatible with almost any types of noun. In example (5), 'linguist' can be replaced by any other types of noun such as 'stadium', 'butterfly', or 'conference' without giving rise to semantic anomaly. These cases are annotated as UNRESTRICTED. Our dataset contains 82 such sentences.

(5) Perhaps you just need a **linguist**. (lt: *human*, ct: $\emptyset$)

An overview of the annotation labels and their relation with the contextual type is summarized as Table 1.

	matching	coercion	other_mis.	unrestricted
Types	$lt = ct$	$lt \neq ct$	$lt \neq ct$	$ct = \emptyset$
#	1410	81	16	82

Table 1: The meanings of the annotation labels and the number of sentences with the labels

4 Method

Based on the annotated dataset, we construct graphs in which each node corresponds to an annotated instance of a noun. Edges are established according to the similarity between the embeddings of the instances. For each node, we examine its neighboring nodes with respect to their semantic types and propose quantitative measures to characterize the distribution of neighbor types. By comparing these measures across instances belonging to different sentence categories (i.e. MATCHING, COERCION, OTHER_MISMATCH, UNRESTRICTED), we assess the extent to which the relation between the lexical type and the contextual type of an instance is reflected in the graphs.

4.1 Models

We experiment with the contextualized word embeddings of two language models: BERT (Devlin et al., 2019) (base, uncased) and sense-enhanced BERT (Yavas et al., 2025). Both models and their embeedings are accessed via Hugging Face[4] and the Transformers library (Wolf et al., 2020). Sense-enhanced BERT is a variant of BERT created by fine-tuning in order to incorporate external semantic knowledge into the model embeddings. It is created by fine-tuning BERT on the SemCor corpus (Miller et al., 1993) with WordNet supersense labels. This model is relevant to our study since supersenses are broad semantic categories that correspond well to semantic types, such as *animal*, *location*, and *state*.

[3]Lexical types are already known from the noun selection process and therefore do not require re-annotation.

[4]Models: *google-bert/bert-base-uncased* and *yavasde/sense-enhanced-bert*. Via: https://huggingface.co/

We extract the embeddings of the target words in the sentences from the last 4 layers of the models and average them to obtain one embedding per target word instance. Averaging the last four layers has been previously used for semantics-related tasks (Liu et al., 2021; Yavas et al., 2025). If the target word is tokenized into subwords by the model tokenizer, we average the embedding of each subtoken before averaging across layers.

To obtain embeddings without lexical type information, we mask the target word in each sentence using the [MASK] token, as shown in (3-b). We extract the embedding of the [MASK] token and use these masked word embeddings for constructing graphs.

4.2 Graph construction

We conduct a graph-based analysis on the embeddings. Concretely, we construct a directed graph $G = (V, E)$ over the instances in the dataset, where $V = \{v_1, \ldots, v_n\}$ is the set of nodes representing the n instances. Each instance is represented by a contextual word embedding $\mathbf{emb}_i$.

To ensure that edges reflect conceptual similarity between different word types rather than lexical overlap, we only form edges between instances of different target words. A directed edge $(v_i, v_j) \in E$ is formed if v_j is among the k closest neighbors of v_i, determined by the cosine similarity between their embeddings, as in:

$sim_{ij} = \cos(\mathbf{emb}(v_i), \mathbf{emb}(v_j))$,

$E = \{(v_i, v_j) | sim_{ij}$ is among the k highest similarities between v_i and other nodes$\}$

In this paper we set $k = 10$.[5]

As introduced in 4.1, we use four different types of embeddings: BERT, sense-enhanced BERT, masked BERT, masked sense-enhanced BERT. The graphs using these four embeddings are referred to as G_b, G_s, G_{mb} and G_{ms}.

4.3 Neighbor type metrics

We examine the out-neighborhood of each instance, $\mathcal{N}^+(v_i)$, in terms of their types in order to evaluate how well the neighbors reflect the lexical type and the contextual type of the instance. As introduced in 3, in our dataset, for each instance v_i, its lexical type $lt_i \in \mathcal{T}$ and contextual type $ct_i \in \mathcal{T} \cup \emptyset$[6] can be inferred from the annotation. For example, the instance 'stadium' has a lexical type (lt) *location* and a contextual type (ct) *human* in (4-a).

We calculate the distributions of semantic types in the neighborhood of each instance using the Neighbor Type Probability (*NTP*). For each type $t \in \mathcal{T}$, $NTP(t, v_i)$ measures the proportion of neighbors of v_i that have type t, where t_j denotes the type of neighbor v_j:

$$NTP(t, v_i) = \frac{|\{v_j \in \mathcal{N}^+(v_i) \mid t_j = t\}|}{k}$$

Using *NTP*, we can determine how much the neighbors of an instance reflect its lexical type lt_i or its contextual type ct_i by calculating $NTP(lt_i, v_i)$ and $NTP(ct_i, v_i)$. We refer to the specific case where *NTP* is computed for the lexical type lt as the Lexical Neighbor Type Matching Ratio ($NTMR_L = NTP(lt, v)$), and for the contextual type ct as the Contextual Neighbor Type Matching Ratio ($NTMR_C = NTP(ct, v)$).

We propose another metric to measure the diversity of the neighbors in terms of their semantic types: Neighbor Type Entropy (*NTE*). *NTE* calculates the entropy of *NTP* distribution. We define the Neighbor Type Entropy *NTE* as:

$$NTE(v_i) = -\sum_{t \in \mathcal{T}} NTP(t, v_i) \times \log(NTP(t, v_i))$$

For an instance v_i, if most of its neighbors have the same type, the *NTE* value is relatively low; if the distribution of the types of its neighbors are diverse, the *NTE* value is higher.

In general, if the graph is organized based on semantic type relations, the neighbors of an instance should be of the same (lexical or contextual) type or taxonomically related semantic types. Moreover, if the graph highlights the lexical information of the instances, we expect most neighbors of an instance v_i to share the lexical type lt_i, i.e. $NTP(lt_i, v_i) \approx 1$. In contrast, if the graph reflects contextual information more than lexical information, we expect $NTP(ct_i, v_i) \approx 1$.

Furthermore, we expect the values of both *NTP* and *NTE* to vary across different sentence types (MATCHING, MISMATCH and UNRESTRICTED).

[5] Results with other values of k are listed in Table 9 in the Appendix. Since our analysis focuses on relative comparisons across models and sentence types, we choose $k = 10$ as a representative for our research. The impact of k on the graph and the metrics will be investigated in future work.

[6] Theoretically, the contextual type could be a type outside our selected set of type $\mathcal{T}$, but this kind of sentence does not exist in our dataset.

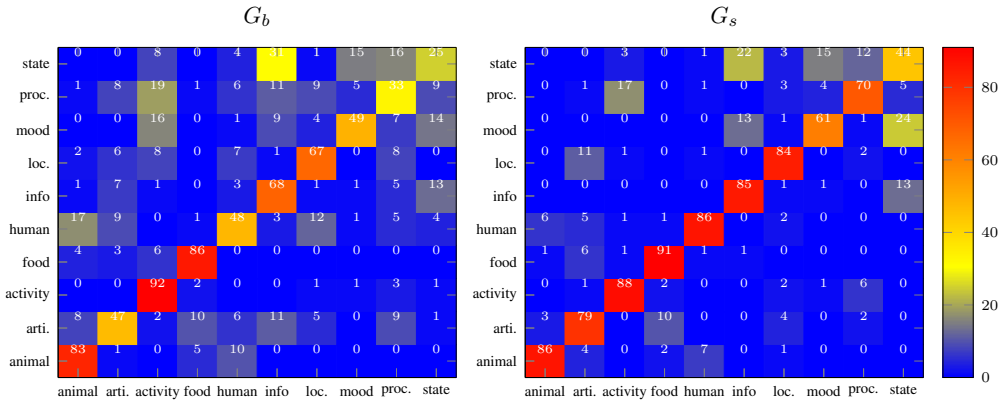


Figure 2: Neighbor Type Ratio (*NTP*) for MATCHING sentences per semantic type, from the graphs based on BERT (G_b) (left) and sense-enhanced BERT embeddings (G_s) (right). The diagonal grids correspond to the average Lexical Neighbor Type Matching Ratio ($NTMR_L$) for each type.

Graph	Sent. type	$NTMR_L$	$NTMR_C$	*other*
G_b	Matching	0.63	—	0.37
	Coercion	0.50	0.17	0.33
	Other_m.	0.54	0.23	0.23
	Unrestr.	0.49	—	0.51
G_{mb}	Matching	0.39	—	0.61
	Coercion	0.18	0.36	0.46
	Other_m.	0.09	0.47	0.44
	Unrestr.	0.14	—	0.86
G_s	Matching	0.81	—	0.19
	Coercion	0.65	0.18	0.17
	Other_m.	0.50	0.41	0.08
	Unrestr.	0.65	—	0.35
G_{ms}	Matching	0.44	—	0.56
	Coercion	0.20	0.40	0.40
	Other_m.	0.11	0.55	0.34
	Unrestr.	0.18	—	0.82

Table 2: Average Lexical Neighbor Type Matching Ratio ($NTMR_L$) and Contextual Neighbor Type Matching Ratio ($NTMR_C$) and the proportion of neighbor types other than lt and ct ($other$) of different sentence types.

These values are obtained by averaging *NTP* and *NTE* values of instances belonging to each sentence type. As to the lexical types in MATCHING sentences, since we avoided polysemous uses of the selected nouns in our dataset, the lexical types of the nouns remain stable. Thus, they are expected to be reliably reflected by the lexical types of the neighbors in the graph. More specifically, the $NTMR_L$ of an instance is expected to be high, and the *NTE* low.

As to MISMATCH sentences, the lexical type of the instance is different from the contextual type. Given that the majority of the instances in our dataset are MATCHING sentences, we expect the types of the neighbors of the instance to be balanced between the two possible options. In this case $NTMR_C$ is likely to be lower and *NTE* slightly higher. For the instances UNRESTRICTED sentences, the contextual types are unclear, so we expect the types of their neighbors to be more diverse. In this case, the $NTMR_L$ should be lower compared to the MATCHING sentences and the *NTE* should be the highest among all sentence categories.

5 Result analysis

5.1 MATCHING sentences

The Neighbor Type Probability (*NTP*) is calculated for each instance. Table 2 reports the average *NTMR* of the instances across different graphs and sentences types. Since G_{mb} and G_{ms} are constructed to capture the information about contextual types instead of lexical types, the $NTMR_L$ on these graphs are significantly lower than on G_b and G_s. Therefore, for MATCHING sentences, our analysis focuses on comparing G_b and G_s.

Fig. 2 presents the average *NTP* of instances in each lexical type in MATCHING sentences.[7] Both heatmaps indicate that the lexical types of most neighbors coincide with those of the instances themselves. Specifically, on G_b, instances of types *animal*, *activity* and *food* exhibit an average $NTMR_L$ over 80%. This proportion is relatively lower for instances of *human*, *artifact* and *mood*, where fewer than half of their neighbors share the same lexical types. In contrast, on G_s, the average $NTMR_L$ increases for almost all types, most notably for types *artifact* and *human*. These observations suggest that G_s reflects the lexical types of the instances more faithfully than G_b.

A closer investigation of the *NTP* for *human* instances further illustrates the increase of $NTMR_L$

[7] More detailed results are given in the appendix.

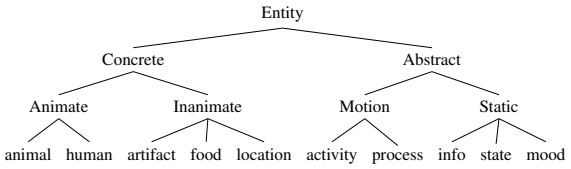


Figure 3: A new type hierarchy induced from the *NTP* values of the semantic types.

Word	G_b	G_s
linguist	80.6	100.0
programmer	48.1	100.0
student	9.3	100.0
genius	54.0	99.3
lady	40.6	99.38
terrorist	50.6	92.6
teenager	88.6	100.0
coward	63.0	81.5
passenger	15.0	43.7
german	86.6	91.6

Table 3: Average Lexical Neighbor Type Matching Ratio (*NTMR*$_L$) of *human* type words on G_b and G_s

on G_s. According to Table 3, certain words, such as 'student' and 'passenger', have very low *NTMR*$_L$ values on G_b. In particular, most neighbors of 'student' instances are of type *location* (predominantly 'classroom' and 'school'), while most neighbors of 'passenger' instances are instances of *artifact* (mainly 'bus' and 'truck'). This indicates that G_b reflects not only semantic type of nouns but also co-occurrence information, whereas on G_s, the semantic type information is more prominently highlighted.

The *NTP* computed on G_s also reveals the distances between semantic types. As shown in Fig. 2, the *NTP* between certain pairs of types is relatively higher than between others, indicating their taxonomic proximity. For example, the type *artifact* is close to *food* and *location*; *activity* is close to *process*; *process*, *state* and *mood* are closer to each other; and interestingly, *info* is close to *mood*. Although this is less common in linguistic studies on type hierarchies, our result suggests a potential prominence in the distinction of abstract and concrete entities. Although this observation requires further investigation, based on the observed *NTP* patterns, we can build a new semantic type hierarchy, presented in Fig. 3.

Furthermore, the *NTP* calculated on G_s can also suggest the types of some non-typical instances. For example, the ontological status of 'alien' and 'robot' is not straightforward. Using our method, according to Table 4, it can be inferred that *alien* is a non-typical member of the type *human*, and *robot* is a non-typical member of *artifact*, and they are close to each other and both similar to the type *animal*.

	alien	robot	anim.	arti.	hum.	other
alien	/	0.27	0.28	0	0.39	0.06
robot	0.39	/	0.24	0.34	0	0.03

Table 4: The neighboring words or types of 'alien' and 'robot' and their distributions.

5.2 UNRESTRICTED sentences

For reasons similar to those observed for MATCHING sentences, the *NTMR*$_L$ values of instances in UNRESTRICTED sentences are higher when computed with unmasked models and with sense-enhanced models. Compared with MATCHING sentences, however, the *NTMR*$_L$ values in UNRESTRICTED sentences are lower across all four graphs. The decrease is more significant with masked models, suggesting that the greater diversity of possible contextual types in UNRESTRICTED sentences is reflected more clearly on G_{mb} and G_{ms}.

The *NTE* further captures the distinction between MATCHING and UNRESTRICTED sentences. Across all four graphs, the diversity of neighbor type distributions is significantly higher for UNRESTRICTED sentences than for MATCHING sentences.

5.3 MISMATCH sentences

The results in Table 2 show that across all graphs, the *NTMR*$_L$ values of MISMATCH sentences (both COERCION and OTHER_MISMATCH) are lower than those of MATCHING sentences, with a considerable proportion of neighbors shifting to the contextual type, as reflected by *NTMR*$_C$. This indicates that the different contextual types exert a noticeable influence on the organization of the graphs.

As discussed in the section on MATCHING sentences, the graphs G_b and G_s largely reflect the information about lexical types. On these graphs, most neighbors of the instances in COERCION sentences share the same lexical types instead of the contextual types. This shows that the embeddings of the instances do not capture this change in the types. This finding seems to suggest that instances in COERCION sentences retain their lexical types and do not undergo a type shift. A more in-depth research is needed to confirm this observation.

G	mat.	coer.	other.	unres.	Comparison
G_b	0.96	1.14	0.99	1.16	mat. $<^{**}$ coer. mat. $<^{**}$ unres. coer. $\not<$ unres. coer. $\neq$ other.
G_{mb}	1.71	1.67	1.64	1.96	mat. $\not<$ coer. mat. $<^{***}$ unres. coer. $<^{*}$ unres. coer. $\neq$ other.
G_s	0.61	0.85	0.65	0.83	mat. $<^{***}$ coer. mat. $<^{***}$ unres. coer. $\not<$ unres. coercion $\neq$ other.
G_{ms}	1.45	1.49	1.42	1.77	mat. $\not<$ coer. mat. $<^{***}$ unres. coer. $<^{**}$ unres. coer. $\neq$ other.

Table 5: Mean Neighbor Type Entropy (*NTE*) of different sentence types and the statistical comparisons between pairs (Mann-Whitney U test) . $^{*}p < .05$, $^{**}p < .01$, $^{***}p < .001$. $\not<$ indicates the hypothesis was not confirmed. The sentence types MATCHING, COERCION, OTHER_MISMATCH and UNRESTRICTED are shortened as MAT., COER., OTHER. and UNRES. respectively.

	NTE on G_b/G_s	*NTE* on G_{mb}/G_{ms}
matching	low	low
coercion	high	low
unrestr.	high	high

Table 6: The relative numerical relations of Neighbor Type Entropy (*NTE*) values among different types of sentences.

With respect to $NTMR_C$, the graphs constructed with masked models produce higher values than the others. As noted above, in the graphs constructed with unmasked models, especially with sense-enhanced BERT, the neighbors of an instance usually have the same lexical type. Consequently, the contextual types of MISMATCH sentences are underrepresented in these graphs, resulting in $NTMR_C < NTMR_L$.

Table 5 reveals that no significant difference is found between COERCION and OTHER_MISMATCH sentences. Therefore, due to the higher number of instances in the dataset, we mainly focus the analysis of the result on COERCION sentences.

A comparison of *NTE* between COERCION and UNRESTRICTED sentences reveals a pattern distinct from that observed between MATCHING and COERCION sentences. On G_{mb} and G_{ms}, the *NTE* of COERCION sentences is significantly lower than that of UNRESTRICTED sentences, whereas on G_b and G_s, no significant difference is found between the two sentence types. This implies that contextual type information is reflected on G_{mb} and G_{ms} to some extent.

For G_{mb} and G_{ms}, although $NTMR_C$ is higher than $NTMR_L$, the average value remains lower than 0.5, indicating that the information about contextual types is not highly prominent in these graphs. This observation is further supported by the analysis of *NTE*. According to Table 5, the *NTE* of instances in MATCHING sentences is significantly lower than the *NTE* of COERCION sentences on G_b and G_s, but not on G_{mb} and G_{ms}. This suggests that G_{mb} and G_{ms} only partially reflect the information about contextual types. We observe that the graphs sometimes capture other information beyond semantic types, such as morphological clues and co-occurrence patterns. For example, on G_{mb} and G_{ms}, the neighbors of the instances of 'elephant' are often instances of 'omelette' and 'explosion', possibly due to the fact that they are all followed by the article 'an'.

Moreover, the information of lexical types is not completely eliminated from G_{mb} and G_{ms}. In these two graphs, the *NTP* of the lexical types are still higher than the *NTP* of other types. This suggests that the lexical information may persist in COERCION sentences. For example, in example (2), although 'finish' strongly indicates a contextual type *activity*, the other words in the sentence, such as 'gulp', still implies the existence of an *artifact* bottle.

The NTP and NTE patterns might also be used to discover possible coercion instances. For example, (6) is a sentence in the data. Since the noun 'pizza' is followed by the preposition 'for', which can normally combine with *food* nouns, usually this sentence is not seen as a COERCION instance. However, with our method, the neighbors of this instance on G_b and G_s are always *food*, while its neighbors on G_{mb} and G_{ms} are always *activity*, implying that other parts of the sentence, such as 'take out', indicate a different contextual type. This example highlights how this method can provide insights into the theoretical distinction between COERCION and literal sentences.

(6) Are you telling me most women you take out for *pizza* don't require bodyguard services?

In summary, the three types of sentences and their relations with *NTE* values across the four

graphs are summarized as Table 6. These results indicate that *NTE* values can potentially help in detecting coercion instances.

6 Conclusive remarks

This paper has presented a graph-based analysis of the contextualized word embeddings of noun instances, examining how semantic type information is reflected across different sentence types. We selected nouns from ten semantic types, extracted corpus instances for each noun, and annotated them with lexical types and contextual types. Instances in which the lexical type coincides with the contextual type are labeled as MATCHING; instances where the two diverge are labeled as MISMATCH, further classified as COERCION or OTHER_MISMATCH; and instances occurring in uninformative contexts are labeled as UNRESTRICTED.

Using these annotated instances, we constructed graphs based on the similarity between their embeddings and proposed two graph-based metrics in order to analyze the neighbors of instances based on their semantic types: Neighbor Type Probability (*NTP*) and Neighbor Type Entropy (*NTE*). Our results demonstrate that graphs constructed with sense-enhanced BERT embeddings reflect information about semantic types better than BERT embeddings. Lexical type information is reliably reflected in graphs constructed on sense-enhanced BERT embeddings, and contextual type information is also partially reflected in graphs constructed on masked embeddings. MATCHING, MISMATCH, and UNRESTRICTED sentences can be distinguished from one another through the comparison of *NTP* and *NTE* values across the graphs.

Our method also has a number of potential future applications. On the linguistic side, the distinction between coercion and other mechanisms underlying lexical-contextual type mismatch can be further explored, given sufficient number of instances with metaphor, metonymy or other kinds of mismatch. A preliminary new type hierarchy has been implied from our method, and with more selection of nouns and, preferably, larger unannotated data, semantic types and their hierarchical organization can possibly be induced automatically, and potential COERCION instances and patterns can possibly be consistently detected. This would also help in refining the definition and diagnostics of semantic types and coercion. Also, our method could be used for polysemous nouns, especially dot types, where the noun has more than one possible lexical types, and might be targeted at the same time on one instance. The graph method might be able to reveal the semantic structure of dot type nouns from a computational perspective.

On the computational side, our graph-based method can serve as a tool for understanding how contextualized word embeddings encode information, with potential extensions to other models and semantic phenomena related to semantic types.

Acknowledgments

This work is part of the project "Coercion and Copredication as Flexible Frame Composition" funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), project number 440934416. We would like to thank Laura Kallmeyer, Rainer Osswald, and the anonymous reviewers for their valuable comments.

References

Nicholas Asher. 2011. *Lexical meaning in context: A web of words*. Cambridge University Press.

Nicholas Asher. 2015. Types, meanings and coercions in lexical semantics. *Lingua*, 157:66–82.

Nicholas Asher, Tim Van de Cruys, Antoine Bride, and Márta Abrusán. 2016. Integrating type theory and distributional semantics: A case study on adjective–noun compositions. *Computational Linguistics*, 42(4):703–725.

Long Chen, Laura Kallmeyer, and Rainer Osswald. 2022. A frame-based model of inherent polysemy, copredication and argument coercion. In *Proceedings of the Workshop on Cognitive Aspects of the Lexicon*, pages 58–67, Taipei, Taiwan. 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.

Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT press.

Frederick Gietz and Barend Beekhuizen. 2022. Remodelling complement coercion interpretation. In *Proceedings of the Society for Computation in Linguistics 2022*, pages 158–170, online. Association for Computational Linguistics.

Qianchu Liu, Fangyu Liu, Nigel Collier, Anna Korhonen, and Ivan Vulić. 2021. MirrorWiC: On eliciting word-in-context representations from pretrained language models. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 562–574, Online. Association for Computational Linguistics.

Zhaohui Luo. 2012. Formal semantics in modern type theories with coercive subtyping. *Linguistics and Philosophy*, 35(6):491–513.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Richard Montague. 1970. Universal grammar. *Theoria*, 36(3):373–398.

James Pustejovsky. 1993. Type coercion and lexical selection. In *Semantics and the Lexicon*, pages 73–94. Springer.

James Pustejovsky. 1995. *The Generative Lexicon*. MIT Press.

Matteo Radaelli, Emmanuele Chersoni, Alessandro Lenci, and Giosuè Baggio. 2025a. 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*, pages 469–480, Vienna, Austria. Association for Computational Linguistics.

Matteo Radaelli, Emmanuele Chersoni, Alessandro Lenci, and Giosuè Baggio. 2025b. Context effects on the interpretation of complement coercion: A comparative study with language models in Norwegian. In *Proceedings of the 16th International Conference on Computational Semantics*, pages 78–88, Düsseldorf, Germany. Association for Computational Linguistics.

Giulia Rambelli, Emmanuele Chersoni, Alessandro Lenci, Philippe Blache, and Chu-Ren Huang. 2020. Comparing probabilistic, distributional and transformer-based models on logical metonymy interpretation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 224–234, Suzhou, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Deniz Ekin Yavas, Timothée Bernard, Benoit Crabbé, and Laura Kallmeyer. 2025. On the relation between fine-tuning, topological properties, and task performance in sense-enhanced embeddings. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23610–23625, Vienna, Austria. Association for Computational Linguistics.

Deniz Ekin Yavas, Laura Kallmeyer, Rainer Osswald, Elisabetta Jezek, Marta Ricchiardi, and Long Chen. 2023. Identifying semantic argument types in predication and copredication contexts: A zero-shot cross-lingual approach. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 310–320, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

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.

Mengjie Zhao, Philipp Dufter, Yadollah Yaghoobzadeh, and Hinrich Schütze. 2020. Quantifying the contextualization of word representations with semantic class probing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1219–1234, Online. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Appendix

Semantic Type	Word	G_b	G_s
info	message	63.75	100.00
	information	88.75	90.00
	poem	13.13	100.00
	rumor	88.75	100.00
	data	81.33	90.00
	news	100.00	100.00
	idea	93.75	96.88
	concept	89.38	100.00
	science	23.75	61.25
	knowledge	78.13	67.50
	problem	10.63	20.63
	secret	81.88	99.38
mood	happiness	31.88	70.00
	empathy	76.88	66.88
	resentment	86.25	97.50
	preference	46.25	75.00
	attention	23.13	13.13
	mood	43.75	65.63
animal	elephant	87.33	99.33
	dinosaur	97.50	96.88
	pig	82.00	79.33
	cat	100.00	98.46
	cow	67.33	84.00
	butterfly	100.00	100.00
	penguin	100.00	93.85
	pigeon	99.33	94.00
	mammal	99.38	100.00
	bug	90.91	96.36
location	cave	99.38	98.75
	classroom	58.75	87.50
	stadium	37.50	93.75
	school	88.75	89.38
	river	88.75	90.00
	road	43.75	22.50
	valley	100.00	99.38
	desert	95.00	100.00
	sky	11.88	80.63
	park	60.00	95.00
state	death	24.38	24.38
	existence	26.88	79.38
	chaos	21.88	46.25
	difficulty	21.25	13.75
	strength	63.75	91.25
	beginning	16.88	53.13
	intelligence	7.50	20.63

Table 7: The average $NTMR_L$ value of each noun per semantic type on G_b and G_s

Semantic Type	Word	G_b	G_s
activity	picnic	88.00	91.33
	barbecue	71.88	61.88
	banquet	90.63	90.63
	celebration	83.75	75.00
	festival	98.13	97.50
	carnival	96.00	94.00
	conference	99.38	90.00
	meeting	98.75	97.50
	parade	86.67	77.33
	party	90.63	85.00
process	explosion	56.25	99.38
	accident	74.38	85.63
	experiment	12.50	78.13
	installation	28.13	90.63
	interruption	15.00	87.50
	storm	20.63	68.13
	sleep	0.00	12.67
	dance	16.25	17.50
	attack	65.00	95.63
	performance	23.13	53.75
food	pizza	91.88	88.75
	kebab	100.00	100.00
	rice	97.50	100.00
	omelette	100.00	100.00
	pork	86.88	95.63
	beef	100.00	100.00
	bread	98.75	100.00
	cake	86.88	100.00
	candy	73.57	92.86
	cigarette	16.88	10.63
	apple	93.57	100.00
	carrot	84.00	96.67
	potato	100.00	100.00
artifact	bus	90.00	100.00
	truck	68.75	100.00
	bicycle	81.88	100.00
	rocket	12.14	58.57
	guitar	43.75	93.75
	sculpture	1.25	71.25
	computer	1.88	72.50
	shirt	80.63	97.50
	coin	25.63	48.13
	cable	43.75	85.63
	button	77.50	99.38
	bottle	69.38	68.13
human	linguist	80.63	100.00
	programmer	48.13	100.00
	student	9.38	100.00
	genius	54.00	99.33
	lady	40.63	99.38
	terrorist	50.67	92.67
	teenager	88.67	100.00
	coward	63.08	81.54
	passenger	15.00	43.75
	german	86.67	91.67

Table 8: The average $NTMR_L$ value of each noun per semantic type on G_b and G_s

Graph	Sent. Type	$k = 10$			$k = 25$			$k = 50$		
		NTMR$_L$	***NTMR***$_C$	*other*	***NTMR***$_L$	***NTMR***$_C$	*other*	***NTMR***$_L$	***NTMR***$_C$	*other*
G_b	MATCHING	0.62	—	0.38	0.60	—	0.40	0.60	—	0.40
	COERCION	0.54	0.18	0.28	0.53	0.20	0.27	0.47	0.23	0.30
	UNRESTR.	0.58	—	0.42	0.50	—	0.50	0.44	—	0.56
G_{mb}	MATCHING	0.37	—	0.63	0.36	—	0.64	0.32	—	0.68
	COERCION	0.28	0.23	0.49	0.17	0.38	0.44	0.17	0.36	0.46
	UNRESTR.	0.26	—	0.74	0.15	—	0.85	0.15	—	0.85
G_s	MATCHING	0.80	—	0.20	0.79	—	0.21	0.76	—	0.24
	COERCION	0.71	0.17	0.12	0.71	0.18	0.10	0.67	0.22	0.11
	UNRESTR.	0.80	—	0.20	0.69	—	0.31	0.67	—	0.33
G_{ms}	MATCHING	0.42	—	0.58	0.42	—	0.58	0.40	—	0.60
	COERCION	0.34	0.23	0.44	0.20	0.41	0.39	0.20	0.40	0.40
	UNRESTR.	0.30	—	0.70	0.20	—	0.80	0.18	—	0.82

Table 9: *NTP* values with different k values on different graphs across sentence types. *NTMR*$_C$ is not applicable (—) for MATCHING and UNRESTRICTED sentences. The results show that on graphs built with unmasked models, the results are relatively stable. On graphs built with masked models, the proportion of neighbors with context types increases significantly when k is set to 25 or 50, and the proportion of neighbors with lexical types decreases accordingly.

Author Index