BriGap 2025

The Second Workshop on the Bridges and Gaps between Formal and Computational Linguistics

Proceedings of the Workshop

September 24, 2025

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

Funded 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 979-8-89176-317-3

Introduction

We are excited to welcome you to BriGap-2, co-located with IWCS 2025, in Düsseldorf, Germany!

This second edition of the workshop on Bridges and Gaps between Formal and Computational Linguistics follows up on the first edition in 2022. We have implemented a number of changes that we hope will reflect the diverse communities that we aim to bring together in this event. In particular, we have worked towards designing an inclusive and welcoming submission policy, soliciting archival papers to be published in the ACL Anthology, non-archival abstracts describing work in progress, as well as presentations of already published articles that would be of interest to our audience.
We are especially proud of the success encountered in this second edition, with 11 publications presented at the workshop, 10 of which are included in these proceedings. The works range across a number of topics, including but not limited to Dependent Type Semantics, the syntactic abilities of LLMs, Lexical Functional Grammer, as well as the use NLP systems for cognitive science. We hope that future editions of the workshop will be able to build upon this success and continue to foster the diversity of topics addressed.
Beyond these 11 presentations, the workshop also includes two invited talks. Anna Rogers (ITU Copenhagen) will discuss data contamination in the age of LLMs, whereas Kees van Deemter (University of Utrecht) will address hallucinations and how to classify them. Both keynotes provide valuable perspectives on pitfalls and caveats of modern NLP technology, and provide an excellent starting point for a broader discussion on how to build successful interactions between formal and computational linguistics.

The BriGap-2 workshop 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.
We also want to thank our colleague Grégoire Winterstein, who helped us put together the workshop proposal before withdrawing from the organizing committee due to an excessive workload.

Timothée Bernard, Timothee Mickus, Program Chairs

Organizing Committee

Program Chairs

Timothée Bernard, Université Paris Cité, CNRS, Laboratoire de linguistique formelle
Timothee Mickus, University of Helsinki

Program Committee

Program Chairs

Timothée Bernard, Université Paris Cité
Timothee Mickus, University of Helsinki

Reviewers

Lasha Abzianidze, Pascal Amsili, Carolyn Jane Anderson

Olivier Bonami, Chloé Braud, Canaan Breiss, Tommi Buder-Gröndahl

Maria Copot, Benoit Crabbé

Isabelle Dautriche, Marie-Catherine De Marneffe

Katrin Erk

Richard Futrell

Juan Luis Gastaldi

Aurelie Herbelot

Gene Louis Kim

Dan Lassiter, Alessandro Lenci, Tal Linzen

Koji Mineshima, Teruyuki Mizuno

Denis Paperno, Christopher Potts

Christian Retoré

Kees van Deemter

Guillaume Wisniewski

Olga Zamaraeva

Keynote Talk

Studying Generalization in the Age of Contamination

Anna Rogers
IT University of Copenhagen
2025-09-24 09:30:00 – Room: **Room 3**

Abstract: In the age of Large Language Models, we can no longer be sure that the test data was not observed in training. This talk discusses the main approaches to studying generalization, and presents a new framework for working with controlled test-train splits across linguistically annotated data at scale.

Bio: Anna Rogers is an tenured Associate Professor at the IT University of Copenhagen is one of the foremost experts in Natural Language Processing (NLP). Her expertise ranges from ethics in NLP to frame semantics, and from computational social science to interpretability. Her contribution to the field goes beyond widely acclaimed scientific articles; she has also taken on significant responsibilities within the community, including heading the scientific committee of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023), and more recently taking on the role of co-editor-in-chief for the ACL Rolling Review platform.
Her recent work at the IT University of Copenhagen focuses on understanding large language models from a sociotechnical perspective. This has led her to studying the impact of data on what these models converge to, as well as how to make NLP models more efficient, transparent and reliable.

Keynote Talk

Classifying Hallucinations in Data-Text NLG: Avoiding the Pitfalls

Kees van Demter
Universiteit Utrecht
2025-09-24 14:20:00 – Room: **Room 3**

Abstract: Algorithms that produce textual output can sometimes "hallucinate", producing texts that express information that differs from what is required. In this presentation, I will talk about hallucination in Data-Text NLG, focusing on situations in which the task of the algorithm is to express a known body of information both fully and accurately. Various attempts have been made to clarify the notion of hallucination, and to distinguish between different types of hallucinations that can occur in the above-mentioned situations. I will examine some of these classifications and ask:
(1) Are the existing classifications well defined? (2) How feasible in practice is it to apply these classifications to concrete cases of Data-Text NLG? (This is joint work with Eduardo Calo and Albert Gatt, both at Utrecht University.) (3) How useful are the distinctions that these classifications make, for example for determining the seriousness of a hallucination, or for redesigning the NLG algorithm so as to avoid hallucinations? And finally, if time permits (4) What does our investigation tell us about hallucinations in other NLG situations, for instance in Question-Answering?

Bio: Kees van Deemter is an Emeritus Professor at the University of Utrecht, where he has been a major support and proponent of research in computational linguistics since 2018. As a long-standing expert in the area of Natural Language Generation, he focuses on structured inputs and their limits: his work ranges from logic-to-text systems to vagueness in natural language, and from referring expressions to some of the caveats of modern neural NLP systems.
Luckily for us, he has recently written an "autoworkography", so we can point you to his own words, which without a doubt will do a much better job of retracing his steps than we can: https://arxiv.org/abs/2504.04142.

Table of Contents

Natural Language Inference with CCG Parser and Automated Theorem Prover for DTS
Asa Tomita, Mai Matsubara, Hinari Daido and Daisuke Bekki 1

Evaluating The Impact of Stimulus Quality in Investigations of LLM Language Performance
Timothy Pistotti, Jason Brown and Michael J. Witbrock 8

Modal Subordination in Dependent Type Semantics
Aoi Iimura, Teruyuki Mizuno and Daisuke Bekki 15

Exploring Gaps in the APS: Direct Minimal Pair Analysis in LLM Syntactic Assessments
Timothy Pistotti, Jason Brown and Michael J. Witbrock 20

Coordination of Theoretical and Computational Linguistics
Adam Przepiórkowski and Agnieszka Patejuk 26

An instructive implementation of semantic parsing and reasoning using Lexical Functional Grammar
Mark-Matthias Zymla, Kascha Kruschwitz and Paul Zodl 35

Modelling Expectation-based and Memory-based Predictors of Human Reading Times with Syntax-guided Attention
Lukas Mielczarek, Timothée Bernard, Laura Kallmeyer, Katharina Spalek and Benoit Crabbé 52

Syntax-Guided Parameter Efficient Fine-Tuning of Large Language Models
Prasanth 72

On the relative impact of categorical and semantic information on the induction of self-embedding structures
Antoine Venant and Yutaka Suzuki 79

Plural Interpretive Biases: A Comparison Between Human Language Processing and Language Models
Jia Ren 97

Natural Language Inference with CCG Parser and Automated Theorem Prover for DTS

Asa Tomita and Mai Matsubara and Hinari Daido* and Daisuke Bekki
Ochanomizu University
{tomita.asa, matsubara.mai, hinari.daido, bekki}@is.ocha.ac.jp

Abstract

We propose a natural language inference (NLI) system that operates on the principles of compositional semantics. The system integrates *lightblue*, a syntactic and semantic parser grounded in Combinatory Categorial Grammar (CCG) and Dependent Type Semantics (DTS), with Wani, an automated theorem prover for Dependent Type Theory (DTT). A key feature of this system is that each computational step corresponds to a specific theoretical assumption, allowing the system's evaluation to function as a form of hypothesis verification. We evaluate our inference system using the Japanese Semantic Test Suite (JSeM) and demonstrate how error analyses can provide feedback for refining both the system and its underlying linguistic theory.

1 Introduction

With the advancement of Natural Language Processing (NLP), the gap between formal linguistics and computational linguistics has been widening. Historically, computational linguistics was deeply intertwined with theoretical linguistics, serving as a means to implement and empirically verify formal linguistic theories. However, the field's focus has progressively shifted towards engineering-oriented approaches, a trend significantly accelerated by the rise of large language models (LLMs).

While LLMs have achieved impressive performance on a wide range of NLP tasks, including natural language inference (NLI) (Cobbe et al., 2021; Wei et al., 2022), their reasoning processes are largely *associative* rather than formally grounded. As a result, their inferences are not based on whether a hypothesis is formally deduced from given premises. Although their outputs are often plausible, concerns persist regarding the reliability and explainability of their inferential processes.

In contrast, inference systems based on formal linguistic theories (Bos, 2008; Chatzikyriakidis and Luo, 2014; Abzianidze, 2017) can output formal proof diagrams that explicitly detail the steps of syntactic, semantic, and theorem proving analysis. A notable example is `ccg2lambda` (Mineshima et al., 2015; Martínez Gómez et al., 2016), an inference system the syntactic parser of which employs Combinatory Categorial Grammar (CCG; Steedman, 1996, 2000), a lexicalized grammar that associates syntactic and semantic information with lexical entries. It generates higher-order logical forms, which are then processed by the Coq theorem prover (The Coq Development Team, 2021).

Despite its theoretical foundation, `ccg2lambda` has limitations stemming from its bi-LSTM-based syntactic parser (Yoshikawa et al., 2017). As CCG concentrates linguistic information within the lexicon, neural parsers make it difficult to precisely diagnose errors at the lexical level. In practice, correcting parsing errors often requires modifying the treebank and retraining the model, which hinders its utility for empirical theory verification.

We conceptualize inference as the ability to formally deduce the semantic representation of a hypothesis from that of the premises. To address the aforementioned challenge, we propose an inference pipeline (Figure 1) that combines *lightblue* (Bekki and Kawazoe, 2016), a robust syntactic and semantic parser based on CCG and Dependent Type Semantics (DTS; Bekki, 2014; Bekki and Mineshima, 2017), with Wani (Daido and Bekki, 2017), an automated theorem prover for DTS. We evaluate this pipeline along with a detailed error analysis.

2 Theoretical Background

2.1 Combinatory Categorial Grammar (CCG)

CCG is a lexicalized grammar that models syntactic structures through a lexicon and a set of combinatory rules. We adopt CCG as our syntac-

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

Proceedings of the Second Workshop on the Bridges and Gaps between Formal and Computational Linguistics (BriGap-2), pages 1–7

tic framework, because it allows for the explicit encoding of syntactic and semantic information within lexical items, providing a clear and localized representation of linguistic structure. This design is particularly well-suited for computational implementations aimed at the empirical verification of linguistic theories, as parsing errors can often be directly attributed to specific lexical entries, facilitating targeted revision.

2.2 Dependent Type Semantics (DTS)

DTS is a type-theoretical framework for natural language semantics based on Dependent Type Theory (DTT; Martin-Löf, 1984). In DTT, the Curry–Howard correspondence establishes an isomorphism between types as propositions, and between terms as proofs. A key feature of this system is its ability to allow types (propositions) to be dependent on terms (proofs). This property allows DTS to represent propositions (types) that contain a reference to a proof from a preceding discourse, Consequently, it reduces phenomena such as anaphora and presupposition resolution to proof search. Since this proof search mechanism is used to validate inferences from premises to conclusions, DTS provides a unified, proof-theoretic account of meaning. Beyond its handling of anaphora and presupposition resolution, the type-theoretical foundation of DTS also allows for the use of type-checking to ensure the consistency of semantic representations. We will examine this property in detail in Section 3.1.2.

3 Inference Pipeline

3.1 Syntactic/Semantic Parser lightblue

lightblue[1] is a syntactic and semantic parser that integrates CCG-based syntactic parsing with DTS-based semantic composition. The syntactic parsing is grounded in the formalization of Japanese CCG as described in Bekki (2010), and it is capable of generating syntactic structures enriched with detailed syntactic features. Furthermore, lightblue incorporates the anaphora resolution mechanism based on type inference to identify anaphoric relations within discourse. The system also verifies the consistency of the derived semantic representations by performing type checking (Bekki and Sato, 2015).

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

3.1.1 Anaphora and Presupposition Resolution with lightblue

In DTS, type checking is employed to verify whether a semantic representation, obtained through semantic composition, is of type `type` in DTT. This condition is referred to as the Semantic Felicity Condition (SFC). Consequently, this process retrieves the contexts that are available for resolving anaphora and presuppositions.

Pronouns and presupposition triggers introduce underspecified terms into the semantic representations. Following semantic composition, lightblue performs type checking, in which each underspecified type launches a proof search, and the Wani system calculates a corresponding (possibly empty) set of proof terms.

The proof terms derived from the above process are used to rewrite underspecified terms, resulting in fully specified semantic representations. This is the first system to implement anaphora and presupposition resolution in DTS, according to its theoretical formulation. This implementation was made possible by the seamless integration of lightblue and Wani.

3.1.2 Type Checking for Evaluating Semantic Analysis

In CCG, semantic composition is derived from the syntactic structure through a homomorphic mapping. Consequently, any ill-formedness in the resulting semantic representations indicates an inconsistency in the corresponding lexical entries. Therefore, the failure of the SFC, as described in the previous section, directly points to an error in some semantic representation specified in the lexicon. In this way, type checking serves as a valuable tool for verifying the internal consistency of the overall implementation.

3.2 Automated Theorem Prover Wani

Wani (Daido and Bekki, 2017) is an automated theorem prover designed for a specific fragment of DTS. Given a set of premises and a conclusion, both formulated as propositions in DTT, Wani attempts to construct a proof. If a proof is found, it outputs the corresponding DTT proof diagram.

Wani performs proof search by applying DTT inference rules to the premises and the conclusion. It combines forward reasoning and backward reasoning strategies. Forward reasoning proceeds from the premises, applying elimination rules to derive

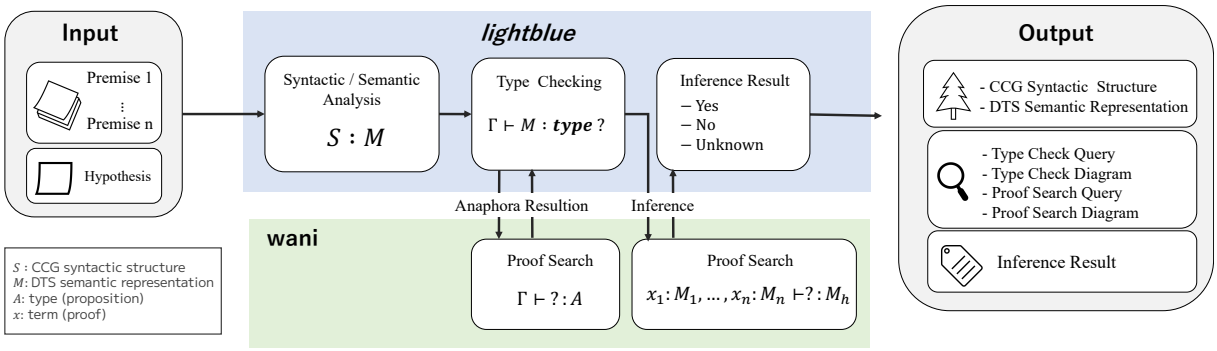


Figure 1: NLI pipeline with lightblue and Wani

their consequences and expand the set of available propositions. In contrast, backward reasoning starts from the conclusion, iteratively working backward to identify the propositions required to apply the rules that would derive it. Wani implements backward reasoning as a depth-first search.

Proof search in DTT is known to be undecidable. To ensure the practical feasibility of Wani, we introduced the following constraints on the search process:

Time and Depth Limits To prevent nontermination, we implemented upper bounds on both the computation time and the number of backward inference steps (i.e., depth). The search is terminated if either of these thresholds is exceeded.

Forward vs. Backward Reasoning While forward reasoning is less flexible, it guarantees termination for elimination rules. Conversely, backward reasoning, while more versatile, can lead to nontermination. To balance these trade-offs, Wani uses forward reasoning for the elimination rule of Σ types and for both the introduction and elimination rules of identity types. All other inference rules are handled via backward reasoning.

Pruning We applied branch pruning to specific backward inference rules for to enhance search efficiency.

3.3 Pipeline Design

The pipeline of the natural language inference system, which utilize lightblue and Wani, is depicted in Figure 1. The process take a set of n-premise sentences and one hypothesis sentence as input, which are then passed to lightblue. For each sentence, lightblue performs syntactic and semantic analyses to compose a semantic representation. A subsequent type-checking procedure is then applied to each semantic representation to ensure that it has the type `type`. At this stage, a context for anaphora and presupposition resolution is incrementally constructed by sequentially adding previously type-checked (fully-specified) semantic representations. This allows type-checking to serve as both a consistency check and a mechanism for anaphora and presupposition resolution. Once all sentences have successfully passed the type-checking phase, Wani is called upon to conduct a proof search. During the search, Wani attempts to construct a proof term of type M_h (the semantic representation of the hypothesis) from the semantic representations of the premise sentences $M_1, \ldots, M_n$. If a proof term is found, Wani returns a proof diagram as output. Based on the output from Wani, lightblue assigns one of three inference labels:

`yes`: A proof term of type M_h is constructed (i.e., the hypothesis is entailed)

`no`: A proof term of type $\neg M_h$ is constructed (i.e., a contradiction)

`unknown`: No proof term is constructed.

Finally, lightblue provides a structured output containing the following information:

Syntactic Structures / Semantic Representations The CCG syntactic structures and DTS semantic composition for the premises $T_1, \ldots, T_n$ and the hypothesis H

Type Checking Information Type checking queries and the corresponding proof diagrams

Proof Search Information Proof search queries and the corresponding proof diagrams

Inference Result The inference label assigned by the system.

JSeM ID:693, Answer:Yes
Premise
ITEL-wa 1993-nen-ni MTALK-o tsukut-ta.
ITEL-TOP 1993-year-in MTALK-ACC
make-PST
(ITEL made MTALK in 1993.)
Hypothesis
ITEL-wa 1993-nen-ni MTALK-o tsukuri-oe-ta.
ITEL-TOP 1993-year-in MTALK-ACC
make-finish-PST
(ITEL finished making MTALK in 1993.)
JSeM ID:703, Answer:Unknown
Premise
Taro-ga Hanako-o sikat-ta.
Taro-NOM Hanako-ACC scold-PST
(Taro scolded Hanako.)
Hypothesis
Taro-ga sikara-re-ta.
Taro-NOM scold-PASSIVE-PST
(Taro was scolded.)

Table 1: Examples in the JSeM dataset

4 Evaluation Experiment

4.1 Dataset: JSeM

The evaluation was conducted on the JSeM dataset (Kawazoe et al., 2015)[2], an inference dataset for Japanese. The dataset contains a mixed set of inference problems: some are direct translations of the English FraCaS test suite (Cooper et al., 1996), while others are specifically designed to address semantic phenomena unique to Japanese. Each problem consists of a set of premises, a hypothesis, and an inference label (yes, no, unknown, or undef, which denotes unacceptable sentences). The problems are further organized into sections categorized in accordance with linguistic phenomena. Examples of data labeled as yes and unknown are shown in Table 1.

4.2 Experiment Setup

We evaluate our system on the 36 problems from the "Verbs" section of JSeM dataset (see Table 1 for examples). This section was selected as it represents the most basic subset of inference problems.

We report the following evaluation metrics: parsing success rate, type-checking success rate, and overall accuracy, precision, recall, and F1 scores. The parsing success rate measures the proportion of problems for which a full syntactic and semantic parse was successfully obtained, as this is a prerequisite for inference. The type checking success rate measures the number of cases where the semantic analysis yielded a well-formed and internally consistent semantic representation. Macro averages treat each class equally, while weighted averages reflect the actual label distribution. Given the class imbalance in the dataset, we report both macro- and weighted-averaged scores for a balanced evaluation.

We emphasize that the 36-problem evaluation set was not used for any system tuning. All components, including parsing, semantic composition, and inference, were applied uniformly without task-specific adjustments.

4.3 Result

Results are shown in Table 2. The evaluation set comprises 72 sentences (36 premises and 36 hypotheses), lightblue generated full parsed trees in 65 sentences, achieving a parsing success rate of approximately 90%. When restricting the evaluation to the 52 unique sentences by removing duplicates, the system achieved full parsed trees for 48, corresponding to a 92.3% success rate. Moreover, type checking succeeded for all parsed sentences, indicating that semantic representations obtained from our semantic analysis satisfied the Semantic Felicity Condition (SFC) and were well-formed. The inference component correctly answered 24 out of the 36 problems. Compared to `ccg2lambda`, our system demonstrated superior performance across all evaluation metrics: accuracy, recall, precision, and F1 score.

Although GPT-4o achieves the highest scores on all metrics, these results should be interpreted as reference values rather than a direct comparison. This is because our research aims at transparent and linguistically grounded inference, which contrasts with the black-box nature GPT-4o. In our framework, a prediction is considered correct only if the system can successfully parse the input, assign a consistent semantic representation, and construct a formal proof. From this perspective, predictions made without a derivable proof, such as GPT-4o's "yes" without an explicit reasoning trace, cannot be fully trusted as valid inferences. Thus, our system prioritizes explainability and credibility based on evidence, over mere surface-level agreement with the correct label.

[2] https://github.com/DaisukeBekki/JSeM

System	ccg2lambda	**Our System**	GPT-4o	Majority
Parsing	-	0.90	-	-
Type Check	-	1.0	-	-
Accuracy	0.556	0.667	0.861	0.806
Precision (macro | weighted)	0.250 | 0.806	0.342 | 0.877	0.438 | 0.951	0.201 | 0.806
Recall (macro | weighted)	0.172 | 0.556	0.397 | 0.667	0.349 | 0.861	0.250 | 1.000
F1 (macro | weighted)	0.204 | 0.658	0.319 | 0.700	0.382 | 0.897	0.223 | 0.892

Table 2: Performance comparison with other systems. Among `ccg2lambda` and our system, the higher value for each metric is underlined. The "Majority" baseline, which assignes the most frequent label ("yes") to all the problems, is also included for reference. For GPT-4o model, we set the temperature to 0.7 and the maximum token limit to 1000. The confusion matrix and the precise prompt used for inference are shown in Table 3 and Figure 2 in the Appendix.

4.4 Error Analysis

Out of the 12 problems with incorrect answers, 7 were attributed to the lack of external world knowledge, 2 to current limitations in Wani's proof search, and the remaining 3 to parsing errors.

4.4.1 External world knowledge

An example of an error attributed to a lack of world knowledge is the following problem:[3]

P: ITEL owned APCOM from 1988 to 1992.

H: ITEL owned APCOM in 1990.

To Correctly infer the hypothesis from the premise, the system requires temporal world knowledge – that 1990 falls within the range from 1988 to 1992 – which is not explicitly encoded.

Incorporating external knowledge presents a well-known challenge. While several studies have explored integrating knowledge bases into their inference systems (Martínez-Gómez et al., 2017; Yoshikawa et al., 2019), these approaches often involve a trade-off where improving recall can lead to a decrease in precision. Therefore, simply injecting more knowledge into the system is insufficient to increase the number of provable cases.

4.4.2 Parsing Error

In our system, we observed parsing errors related to the interpretation of case marks. A representative example is the following sentence[4]:

P: Taro-wa Jiro-kara Hanako-o
Taro-NOM Jiro-from Hanako-ACC
syookaisa -re -ta
introduce PASSIVE PST
'Taro was introduced to Hanako by Jiro.'

In this case, the parser failed to correctly recognize that *kara* ("from") in the passive construction semantically corresponds to the dative argument in the active counterpart. It is known that *kara*-NP is not fully interchangeable with the dative NP, and is not always licensed as a verbal argument. Consequently, resolving such errors requires a deeper linguistic analysis of selectional restrictions and case-marking behavior for specific verbs, rather than a simple modification or addition of lexical entries for *kara*.

5 Conclusion

This paper has presented a linguistically-grounded natural language inference system, which integrates syntactic parsing, semantic composition, type checking, and proof search. Our proposed pipeline demonstrated improved inference accuracy over existing formal systems.

This system offers a promising avenue for bridging the gap between linguistic theory and large language models. Given that all of its technical components are based on hypotheses from formal linguistics, improvements to the system directly contribute to the refinement of theoretical assumptions. Furthermore, lightblue can serve as a novel tool for verifying the outputs of LLMs, thereby facilitating systematic comparisons between data-driven inferences and theory-driven predictions.

Acknowledgments

This work was supported by JST BOOST, Japan Grant Number JPMJBS2406, JSPS KAKENHI Grant Number JP23H03452, Japan, and JST CREST Grant Number JPMJCR20D2, Japan.

[3]JSeM ID: #698
[4]JSeM ID: #717

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. 2010. *Nihongo-Bunpoo-no Keisiki-Riron - Katuyootaikei, Toogohantyuu, Imigoosei - (trans. 'Formal Japanese Grammar: the conjugation system, categorial syntax, and compositional semantics')*. Kuroshio Publisher, Tokyo.

Daisuke Bekki. 2014. Representing anaphora with dependent types. In *Logical Aspects of Computational Linguistics*, pages 14–29, Berlin, Heidelberg. Springer Berlin Heidelberg.

Daisuke Bekki and Ai Kawazoe. 2016. Implementing variable vectors in a CCG parser. In *Logical Aspects of Computational Linguistics. Celebrating 20 Years of LACL (1996–2016)*, pages 52–67, Berlin, Heidelberg. Springer Berlin Heidelberg.

Daisuke Bekki and Koji Mineshima. 2017. *Context-Passing and Underspecification in Dependent Type Semantics*, pages 11–41. Springer International Publishing, Cham.

Daisuke Bekki and Miho Sato. 2015. Calculating projections via type checking. In *TYpe Theory and LExical Semantics (TYTLES), ESSLLI2015 workshop*.

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.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Josef Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, Steve Pulman, Ted Briscoe, Holger Maier, and Karsten Konrad. 1996. FraCaS: A Framework for Computational Semantics. Technical Report Deliverable D16, FraCaS Consortium.

Hinari Daido and Daisuke Bekki. 2017. Development of an automated theorem prover for the fragment of DTS. In *the 17th International Workshop on Logic and Engineering of Natural Language Semantics (LENLS17)*.

Ai Kawazoe, Ribeka Tanaka, Koji Mineshima, and Daisuke Bekki. 2015. A framework for constructing multilingual inference problem sets: Highlighting similarities and differences in semantic phenomena between English and Japanese. In *MLKRep2015*.

Per Martin-Löf. 1984. *Intuitionistic Type Theory Vol. 1*. Bibliopolis.

Pascual Martínez Gómez, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2016. ccg2lambda: A computational semantics system. In *the Association of Computational Linguistics (ACL2016)*, pages 85–90.

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.

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.

Mark Steedman. 1996. *Surface Structure and Interpretation*. The MIT Press, Cambridge.

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

The Coq Development Team. 2021. The coq reference manual: Release 8.14.1.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

Masashi Yoshikawa, Koji Mineshima, Hiroshi Noji, and Daisuke Bekki. 2019. Combining axiom injection and knowledge base completion for efficient natural language inference. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7410–7417.

Masashi Yoshikawa, Hiroshi Noji, and Yuji Matsumoto. 2017. A* CCG parsing with a supertag and dependency factored model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 277–287, Vancouver, Canada. Association for Computational Linguistics.

A Appendix

		GPT 4o				ccg2lambda				lightblue			
		Yes	No	Unk	Other	Yes	No	Unk	Other	Yes	No	Unk	Other
Ground Truth	Yes	28	0	1	0	20	0	1	8	17	0	12	0
	No	0	0	0	0	0	0	0	0	0	0	0	0
	Unk	0	4	3	0	0	0	0	7	0	0	7	0
	Other	0	0	0	0	0	0	0	0	0	0	0	0

Table 3: Confusion matrix of inference systems

与えられた文のペアについて、文Aが文Bを含意するかどうかを判定し、そのペアに⟨DATASET_LABEL⟩を付けてください。⟨DATASET_LABEL⟩は以下のいずれかです：
yes：前提が仮説を含意する
no：前提が仮説の否定を含意する
unknown：前提が仮説を含意せず、その否定も含意しない
undef：与えられた情報のみからは判断ができない

文A：⟨PREMISE_SENTENCE⟩
文B：⟨HYPOTHESIS_SENTENCE⟩
####
⟨DATASET_LABEL⟩のみ出力してください。

—- English Translation —-
Given a pair of sentences, determine whether Sentence A entails Sentence B, and assign a ⟨DATASET_LABEL⟩ to the pair.⟨DATASET_LABEL⟩ must be one of the following:
yes: the premise entails the hypothesis
no: the premise entails the negation of the hypothesis
unknown: the premise entails neither the hypothesis nor its negation
undef: it is not possible to determine based on the given information alone

Sentence A: ⟨PREMISE_SENTENCE⟩
Sentence B: ⟨HYPOTHESIS_SENTENCE⟩

####
Only output ⟨DATASET_LABEL⟩.

Figure 2: Prompt designed for LLMs to assign the entailment relation label <DATASET_LABEL>, and its English translation

Evaluating The Impact of Stimulus Quality in Investigations of LLM Language Performance

Timothy Pistotti
University of Auckland

Jason Brown
University of Auckland

Michael Witbrock
University of Auckland

Abstract

Recent studies employing Large Language Models (LLMs) to test the Argument from the Poverty of the Stimulus (APS) have yielded contrasting results across syntactic phenomena. This paper investigates the hypothesis that characteristics of the stimuli used in recent studies, including lexical ambiguities and structural complexities, may confound model performance. A methodology is proposed for re-evaluating LLM competence on syntactic prediction, focusing on GPT-2. This involves: 1) establishing a baseline on previously used (both filtered and unfiltered) stimuli, and 2) generating a new, refined dataset using a state-of-the-art (SOTA) generative LLM (Gemini 2.5 Pro Preview) guided by linguistically-informed templates designed to mitigate identified confounds. Our preliminary findings indicate that GPT-2 demonstrates notably improved performance on these refined PG stimuli compared to baselines, suggesting that stimulus quality significantly influences outcomes in surprisal-based evaluations of LLM syntactic competency.

1 Introduction

The Argument from the Poverty of the Stimulus (APS) remains a central topic in linguistics and cognitive science, and proposes that human linguistic competence extends beyond that supported by direct evidence available during acquisition, thereby implying contributions of innate knowledge to language learning (Chomsky, 1980). Using artificial neural networks as proxies for unbiased learners, recent studies have explored the generalizations that Large Language Models (LLMs) form about linguistic phenomena. A promising line of research compares token probabilities in minimal pairs (e.g., (Linzen et al., 2016; Futrell et al., 2019; Wilcox et al., 2024; Lan et al., 2024)) following Elman's 1990 recommendation that language models be treated as human subjects in psycholinguistic studies.

Wilcox et al. (2024) provide significant findings in this area, demonstrating that LLMs can achieve high performance on various English filler-gap dependencies and island constraints, as measured by surprisal metrics applied to critical regions of minimal pairs of sentences (Wilcox et al., 2024). Their results challenge the necessity of linguistic innateness for these particular syntactic structures.

Building on this work, Lan et al. (2024) investigate more complex, lower frequency syntactic constructions, notably parasitic gaps (PGs) and across-the-board (ATB) movement, but argue that the observed failures of LLMs (including GPT-2) to adequately learn these structures support the APS.

This paper limits its scope to the evaluation of PGs in the context of Lan et al.'s 2024 analysis. We argue that while their work addresses crucial linguistic questions, a critical examination of their PG stimuli reveals characteristics that may interfere with LLM performance. These characteristics include: 1) unintended lexical ambiguities, 2) the structural complexity of the noun phrases hosting parasitic gaps, and 3) potential alternative repairs to ungrammaticality.

The central aim of this research is to investigate the extent to which such properties affect an LLM's predictive power in critical regions of PGs. We propose a methodology centred on generating controlled stimuli using a SOTA generative LLM (Gemini 2.5 Pro Preview) guided by precise, linguistically-informed templates. This approach seeks to mitigate the identified potential confounds while allowing for some flexibility in generation. We present preliminary findings, comparing model performance on our dataset to baselines derived from the original Lan et al. (2024) PG data. Our results suggest that stimulus quality has a significant impact on surprisal scores in critical regions, with implications for the broader APS debate and for

Proceedings of the Second Workshop on the Bridges and Gaps between Formal and Computational Linguistics (BriGap-2), pages 8–14

researchers interested in applying surprisal-based methods to the investigation of LLM capabilities.

2 Parasitic Gap Stimuli

Using a Context-Free Grammar (CFG), (Lan et al., 2024, Table 2, p. 16) generated a total of 8,064 sentence tuples each comprised of $\pm Filler$, $\pm Gap$ variations, exemplified in Table 1. While this approach allows for controlled generation, close examination reveals characteristics of the resulting PG stimuli that may influence model performance independently from the core syntactic properties of PG licensing.

2.1 Unintended Ambiguity

This section identifies two ambiguities prevalent in Lan et al.'s PG dataset. A particularly prominent example involves the use of possessive gerunds (e.g., "John's talking") within the subject noun phrase (NP) that hosts the first gap (G1).

(1) *I know who [John's talking to _] is going to annoy you soon.

Here, "John's" is ambiguous between a contraction of "John is" and the possessive "John + GEN". If interpreted as "John is talking to _," the embedded phrase might not form the intended island structure necessary for a PG, or its grammaticality profile changes. Conversely, if interpreted as a possessive, it forms the intended complex NP island. Given that the stimuli presented to the LLMs were unbracketed and unannotated (as confirmed by the project's public repository), the model must disambiguate this string without a forced reading of sentence structure. Similarly, constructions such as "intent to" (e.g., in "I know who Bob's intent to talk to _ is about to bother soon" include the same ambiguity with the addition of a potential alternative rescue for the sentence's overall grammaticality (e.g., "intent on talking to" or "intention of talking to") that might alter processing ease.

2.2 Structural Complexity of Noun Phrases

The parasitic gap (G1) in Lan et al. stimuli is embedded within a subject NP that forms an island, derivable from their CFG rules such as "`(NP_COMPLEX) -> (N_EMBEDDED) 'to' (V_EMBEDDED)`" (Lan et al., 2024, Table 2, p. 16), leading to structures such as the underlined portion of

(2) I know who Bob's decision to dance with _ is likely to bother eventually.

While subject NPs are indeed syntactic islands, the internal complexity of these specific `NP_COMPLEX` structures (involving nominals followed by an infinitival phrase) introduces a degree of structural depth that goes beyond the more canonical adjunct PG constructions often cited as core examples in the literature (Culicover et al., 2001). This complexity might itself be a confounding factor for LLMs.

3 Method

To investigate the impact of stimulus characteristics on LLM performance for PG constructions, an experiment was designed to compare model performance across three datasets: the original Lan et al. stimuli, a filtered version of this original set, and a new, refined set generated for this study. For our analysis, we selected GPT-2 as the primary evaluation model. This choice is motivated by two factors: first, its use in both Wilcox et al. (2024) and Lan et al. (2024) provides a direct point of comparison with prior findings. Second, while GPT-2 possesses sophisticated language capabilities, it precedes the current era of massive-scale models. This makes it a more suitable testcase for hypotheses related to the Argument from the Poverty of the Stimulus, as it is less likely to have encountered rare syntactic constructions, such as parasitic gaps, at a high frequency during its training.

3.1 Evaluation Metric: Surprisal

Our primary measure of model performance is **surprisal**, which quantifies how unexpected a given word (w_i) is in its preceding context (C). Following standard practice (Wilcox et al., 2024; Lan et al., 2024), surprisal is calculated as the negative log probability, in bits:

$$S(w_i \mid C) = -\log_2 P(w_i \mid C) \quad (1)$$

Lower surprisal values indicate that a word is more predictable. We used this metric to calculate the Δ and Difference-in-Differences (DiD) metrics as proposed by Lan et al. (2024), where $\Delta = S(\text{-Gap Continuation}) - S(\text{+Gap Continuation})$. Model success in modelling the relevant grammaticality judgement is indicated by $\Delta_{+\text{filler}} > 0$ and $\text{DiD} = (\Delta_{+\text{filler}} - \Delta_{-\text{filler}}) > 0$.

3.2 Datasets

We compare GPT-2's performance across three distinct datasets for PGs:

Table 1: Example paradigm for parasitic gaps. Underlined words indicate the filler alternations. Boldfaced words indicate the critical region that shows whether the continuation is gapped or not. Reproduced from Table 4 Lan et al. (2024, p. 19).

	+Gap	**-Gap**
+Filler	I know who John's talking to is about to annoy **soon**.	I know who John's talking to is about to annoy **you** soon.
-Filler	I know that John's talking to Mary is about to annoy **soon**.	I know that John's talking to Mary is about to annoy **you** soon.

(a) **Original Lan et al. (2024) Stimuli:** The full dataset generated from their CFG (N=8064 items), extracted from their publicly available materials.

(b) **Filtered Lan et al. (2024) Stimuli:** A subset of the original dataset (N=5760 items) excluding all items containing the specific ambiguous constructions identified in Section 2, namely those following the pattern: "NAME's VERBing to".

(c) **Refined Stimuli (This Work):** A new, controlled dataset of subject PG constructions generated using Gemini 2.5 Pro Preview (see Appendix A for the full prompt template). This generation was guided by precise structural templates designed to mitigate the confounds present in the original dataset, including using unambiguous "the [NounHead] of/about G1" structures for the subject island, ensuring pragmatically plausible coindexation, and using single-word critical regions for the main clause gap (G2) comparison. All such generated items underwent manual review for grammaticality.

3.3 Experimental Procedure

For each dataset, we followed an identical experimental procedure:

(1) **Data Preprocessing:** Stimuli are formatted into a long-format CSV with columns for `sentence_type`, `item_id`, `condition`, and `full_sentence`.

(2) **Surprisal Extraction:** BPE-level surprisals for each sentence are obtained from GPT-2 using a Python pipeline leveraging the `lib.py` framework from Lan et al.'s (2024) repository.

(3) **Critical Region Aggregation:** Surprisals for the single-word critical regions (the overt object NP in '-Gap' conditions or the post-gap adverb in '+Gap' conditions) are calculated by summing the surprisals of their constituent BPEs.

(4) **Analysis:** The Δ and DiD metrics, along with accuracies and one-sample t-tests, are calculated for each dataset to allow for direct comparison.

4 Preliminary Findings and Discussion

Following the methods outlined in Section 3, we conducted a preliminary evaluation using GPT-2. The primary focus was to assess whether refining the stimuli for PG constructions, specifically addressing the potential confounds identified in Lan et al.'s 2024 dataset, would lead to a different pattern of performance for GPT-2.

4.1 GPT-2 Performance on Original, Filtered, and Refined PG Stimuli

Lan et al. (2024) originally reported that GPT-2 performed poorly on PG stimuli, with key metrics around 5.6% accuracy for the $\Delta_{\text{+filler}} > 0$ criterion and 68.8% accuracy for the Difference-in-Differences (DiD) criterion (Lan et al., 2024, Figs. 5 & 6, pp. 18, 21). This was presented as support for APS.

To establish a direct baseline, our pipeline confirmed these findings on the unfiltered original Lan et al. (2024) dataset (N=8064), yielding an accuracy of **5.61%** for $\Delta_{\text{+filler}} > 0$ and **68.75%** for the DiD metric. Next, GPT-2's performance on the filtered subset (N=5760) was analysed. On this filtered set, accuracy for the $\Delta_{\text{+filler}} > 0$ criterion improved to **7.01%** ($\chi^2(1) = 11.2381, p = 0.0008$), and for the DiD criterion, accuracy improved to **72.93%** ($\chi^2(1) = 28.0780, p < 0.0001$). These results provide initial empirical support for the identified constructions acting as confounds.

Finally, GPT-2 was evaluated on newly generated, refined `subject_pg` stimuli (N=10 items). This yielded significant further improvement: for the $\Delta_{\text{+filler}} > 0$ metric, accuracy rose to **60.0%** ($t(9) = 1.66, p = 0.066$, one-tailed), and for the DiD metric, accuracy reached **80.0%**, a statistically significant effect ($t(9) = 2.64, p = 0.013$, one-tailed; 95% CI $[0.39, 5.01]$). These comparative accuracy scores are visualized in Figure 1.

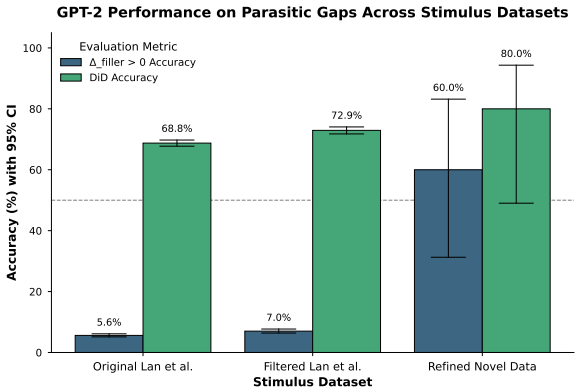


Figure 1: Comparison of GPT-2 accuracy on Parasitic Gap constructions. Accuracy is shown for the $\Delta_{+\text{filler}} > 0$ and Difference-in-Differences (DiD) > 0 criteria across the original (Lan et al., 2024) dataset, a filtered version, and our own refined stimuli. Error bars represent 95% confidence intervals.

4.2 Discussion

The preliminary findings from our refined `subject_pg` dataset indicate a marked improvement in GPT-2's performance compared to the results reported by Lan et al. (2024) for their original PG stimuli using the same model. The DiD accuracy increased from ∼69% to 80%, and notably, the direct preference accuracy ($\Delta_{+\text{filler}} > 0$) jumped from ∼6% to 60%.

While these results are based on an initial set of refined stimuli and a single model, they suggest that characteristics of the test stimuli play a substantial role in LLM evaluations of complex syntax. The reduction of lexical ambiguities (like the "John's" issue) and the use of more canonical island structures for the G1-hosting subject NP may have allowed GPT-2 to better demonstrate any underlying sensitivity it has to PG constructions.

These findings do not nullify Lan et al.'s (2024) broader arguments regarding the APS, that more complex linguistic phenomena may be better suited to test learnability. However, they do suggest that conclusions about an LLM's failure to acquire a phenomenon might be premature if based on stimuli containing significant potential confounds. If an LLM's performance is demonstrably better on refined, unambiguous stimuli, it points to the model's sensitivity to these confounds, and implies that at least some of the previously observed "failure" might be attributable to the nature of the test items themselves rather than to incomplete generalization. This suggests that the introduction of unintended complexities, not directly targeted by the parasitic gap investigation, may obscure an LLM's underlying sensitivity to PG licensing, analogous to the effect of increased structural complexity (e.g., embedding depth) Wilcox et al. (2024) in reducing wh-effects in filler-gap dependencies.

The approach of using a SOTA generative LLM (Gemini 2.5 Pro Preview) guided by precise linguistic templates for creating these refined stimuli shows promise as a method for developing more robust and theoretically sound evaluation protocols. This can help in disentangling true model capabilities from noise introduced by problematic test data.

4.3 Limitations and Future Directions

Future work based on these preliminary findings will involve:

- Expanding the refined dataset to include more items and other PG structures (e.g., adjunct PGs).
- Testing a wider range of LLMs, including more recent architectures and models trained on smaller datasets.
- Conducting a more detailed error analysis on the original Lan et al. (2024) PG dataset using our full pipeline to quantify the impact of specific item characteristics.
- Further refining the LLM-based stimulus generation methods.

5 Conclusion

This work investigated the impact of stimulus quality on the evaluation of LLM knowledge of complex syntax, focusing on parasitic gaps as studied by Lan et al. (2024). Potential confounds in their stimuli were identified, and it was demonstrated that GPT-2's performance on parasitic gap constructions improves significantly when evaluated on a refined dataset designed to mitigate these issues.

Preliminary results suggest that conclusions about an LLM's failure to acquire a phenomenon may be premature if based on stimuli with confounds. This underscores the critical importance of stimulus design. The initial results reported here underscore the critical importance of stimulus quality in the evaluation of LLM syntactic abilities and have direct bearing on debates surrounding linguistic nativism and learnability.

References

Noam Chomsky. 1980. Rules and representations. *Behavioral and brain sciences*, 3(1):1–15.

Peter W Culicover and 1 others. 2001. Parasitic gaps: A history. *Current Studies in Linguistics Series*, 35:3–68.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. 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 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Nur Lan, Emmanuel Chemla, and Roni Katzir. 2024. Large language models and the argument from the poverty of the stimulus. *Linguistic Inquiry*, pages 1–28.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Ethan Gotlieb Wilcox, Richard Futrell, and Roger Levy. 2024. Using computational models to test syntactic learnability. *Linguistic Inquiry*, 55(4):805–848.

A Prompt Template for Stimulus Generation

Prompt for data generation:

```
Your task is to generate 10 unique item sets for testing parasitic gap constructions
    in English. Each item set must consist of exactly four sentences, following a
    $2 \times 2$ factorial design: +/- Filler and +/- Main Clause Gap (G2). The
    output should be formatted as a series of comma-separated lines, with each line
    representing one sentence.

**Objective:**

The primary goal is to create natural-sounding and grammatically clear sentences.
    The `+Filler, +Gap` sentence in each set must be a canonical parasitic gap
    construction where the wh-filler "who" is co-indexed with two gaps: G1 (the
    parasitic gap within a subject NP island) and G2 (the host gap, object of the
    main embedded verb). This co-indexed reading should be pragmatically plausible.
    The critical material differentiating the `+G2` (gapped) and `-G2` (filled)
    conditions for the main clause verb must be a single word.

**Core Sentence Structure for Parasitic Gap (`+Filler, +Gap` condition):**

`[Preamble] who [SubjectNP containing G1] [MatrixVerbPhrase licensing G2] [
    ADV_Post_G2_Gap].` (Note: The gap for G2 is implied before the ADV_Post_G2_Gap).

**Detailed Constraints for Sentence Components:**

1. **Preamble:** Choose from simple introductory phrases like: "I know", "She heard
    ", "They believe", "The report suggested", "It is clear".
2. **Filler/Complementizer:**
* `+Filler` conditions use: "who"
* `-Filler` conditions use: "that"
3. **Subject NP containing G1 (The Island):**
* This NP must be the subject of the matrix verb phrase. The gap G1 is the object of
     the preposition.
* Structure: "the `[NounHead]` `[Preposition]`" (The gap G1 is implied after the
    preposition).
* `[NounHead]`: Use common nouns that naturally take a PP complement with "about" or
     "of" where the object of the preposition can be a person. Examples: "story", "
    report", "book", "article", "picture", "critique", "rumor", "discussion", "
    painting", "description".
* `[Preposition]`: Use **only "about" or "of"**. Select the preposition that forms
    the most natural phrase with your chosen `[NounHead]`.
4. **Matrix Verb Phrase (licensing G2):**
* Structure: `[LinkingVerb] [TransitiveVerb_G2]` (The gap G2 or object
    G2_FillerObject follows this).
* `[LinkingVerb]`: Use common linking phrases like: "is likely to", "is going to", "
    is expected to", "will probably", "might".
* `[TransitiveVerb_G2]`: Use common transitive verbs that naturally take a person as
     a direct object (for G2). Examples: "upset", "amuse", "delight", "interest", "
    surprise", "anger", "please", "concern", "bother", "disturb", "fascinate".
5. **Critical Word for +G2 (Gapped) Condition:**
* `[ADV_Post_G2_Gap]`: When G2 is gapped, the sentence should continue immediately
    after `[TransitiveVerb_G2]` with a single, common adverb from the following list
     ONLY: "soon", "eventually". This adverb signals the gapped G2.
6. **Lexical Items for Filled Gaps:**
* `[G1_FillerName]` (fills G1 in `-Filler` conditions that also have G1 filled): Use
     common, simple proper names (e.g., "Mary", "John", "Sarah", "the manager").
* `[G2_FillerObject]` (fills G2 in `-G2` conditions): **Use ONLY a single common
    proper name from a list such as: "Anna", "Ben", "Chris", "Dana", "Leo", "Sara",
    "Tom", "Paul", "Nina". Please vary the names used. Avoid using "Kim" for this
    slot if other simple names from this list or similar common single names are
    suitable.** The goal is a single-word proper name.
* Ensure `[G1_FillerName]` and `[G2_FillerObject]` are different within the same
    item set.


**Factorial Design - Sentence Patterns for Each Item Set:**
(Note: Gaps are implied by the structure and absence of overt objects.)
```

```
1. **`PFPG` (`+Filler, +G1_gap, +G2_gap`):**
`[Preamble] who the [NounHead] [Preposition] [LinkingVerb] [TransitiveVerb_G2] [
    ADV_Post_G2_Gap].`
*Example: I know who the story about is likely to amuse soon.*
2. **`MFPG` (`-Filler, +G1_filled, +G2_gap`):**
`[Preamble] that the [NounHead] [Preposition] [G1_FillerName] [LinkingVerb] [
    TransitiveVerb_G2] [ADV_Post_G2_Gap].`
*Example: *I know that the story about Mary is likely to amuse soon.*
3. **`PFMG` (`+Filler, +G1_gap, -G2_filled`):**
`[Preamble] who the [NounHead] [Preposition] [LinkingVerb] [TransitiveVerb_G2] [
    G2_FillerObject] [ADV_Post_G2_Gap].`
*Example: *I know who the story about is likely to amuse Anna soon.*
4. **`MFMG` (`-Filler, +G1_filled, -G2_filled`):**
`[Preamble] that the [NounHead] [Preposition] [G1_FillerName] [LinkingVerb] [
    TransitiveVerb_G2] [G2_FillerObject] [ADV_Post_G2_Gap].`
*Example: I know that the story about Mary is likely to amuse Anna soon.*
**Output Format and Instructions for Generation:**

Please provide 10 unique item sets. For each item set, output four lines, each
    corresponding to one of the conditions below. Each line must follow this exact
    comma-separated format:

`sentence_type,item_id,condition,full_sentence`
* **`sentence_type`**: Use the value "subject_pg" for all sentences.
* **`item_id`**: Use a unique integer for each set (e.g., 1 for the first set of
    four sentences, 2 for the second set, and so on, up to 10).
* **`condition`**: Use the labels "PFPG", "MFPG", "PFMG", "MFMG" respectively for
    the four sentences in each item set, corresponding to the patterns defined above
    .
* **`full_sentence`**: The generated sentence string, ending with a period.

**Example of desired output format for ONE item set (item_id 1):**
subject_pg,1,PFPG,I know who the story about is likely to amuse soon.
subject_pg,1,MFPG,I know that the story about Mary is likely to amuse soon.
subject_pg,1,PFMG,I know who the story about is likely to amuse Anna soon.
subject_pg,1,MFMG,I know that the story about Mary is likely to amuse Anna soon.

**Crucial Reminders for Generation:**
* Vary lexical choices for `[Preamble]`, `[NounHead]`, `[Preposition]` (choose 'of'
    or 'about'), `[G1_FillerName]`, `[LinkingVerb]`, `[TransitiveVerb_G2]`, `[
    G2_FillerObject]` (from the restricted list of names), and `[ADV_Post_G2_Gap]` (
    from the restricted list) across the 10 item sets to ensure diversity.
* All `PFPG` sentences must be natural, unambiguously grammatical parasitic gap
    constructions with a pragmatically plausible co-indexed reading for "who". The
    subject NP containing G1 must clearly function as a syntactic island.
* All grammatical sentences (PFPG and MFMG) must be clearly grammatical;
    ungrammatical sentences (MFPG and PFMG) must be clearly ungrammatical due to the
     specified filler/gap violations.
```

Listing 1: Gemini 2.5 Prompt Template

Modal Subordination in Dependent Type Semantics

Aoi Iimura **Teruyuki Mizuno** **Daisuke Bekki**
Ochanomizu University
{iimura.aoi, bekki}@is.ocha.ac.jp
mizuno.teruyuki@ocha.ac.jp

Abstract

In the field of natural language processing, the construction of "linguistic pipelines", which draw on insights from theoretical linguistics, stands in a complementary relationship to the prevailing paradigm of large language models. The rapid development of these pipelines has been fueled by recent advancements, including the emergence of Dependent Type Semantics (DTS) — a type-theoretic framework for natural language semantics. While DTS has been successfully applied to analyze complex linguistic phenomena such as anaphora and presupposition, its capability to account for modal expressions remains an underexplored area. This study aims to address this gap by proposing a framework that extends DTS with modal types. This extension broadens the scope of linguistic phenomena that DTS can account for, including an analysis of modal subordination, where anaphora interacts with modal expressions.

1 Introduction

In recent computational linguistics research, a new approach to natural language processing has seem rapid progress: the use of *linguistic pipelines* (Abzianidze, 2015; Mineshima et al., 2015). These pipelines combine theoretical linguistic insights with computational methods. A key driver of this progress is Dependent Type Semantics (DTS) (Bekki and Mineshima, 2017), a framework for natural language semantics that is rooted in Dependent Type Theory (DTT) (Martin-Löf, 1984). Drawing upon the rich tradition of type theory in programming semantics, DTS provides a compositional framework for the analysis of anaphora and presupposition, which exploits theorem provers in analyzing both anaphora resolution and general inference. By a systematic mapping from formal syntax to semantic interpretation, DTS bridges a significant gap between linguistic theories and computational implementation.

In DTS, the semantic representation (SR) of a sentence corresponds to a type in DTT. The dependency of a type on terms allows reference to terms constructed from the context, thereby reducing both anaphora resolution and presupposition binding to problems of proof search. While DTS provides compelling analyses of complex linguistic phenomena, empirical research on modal expressions remains largely unexplored (but see Tanaka et al. 2015), with existing studies primarily focusing on propositions that abstract away from modal expressions. This study aims to extend DTS by providing an analysis of phenomena involving modal expressions.

Modal expressions, which pertain to the notions of possibility and necessity, have been a central research topic in formal semantics. One of the most discussed phenomena is modal subordination (MS), which, since the pioneering work by Roberts (1989), has been investigated by many researchers (Frank and Kamp, 1997; Kaufmann, 2000; van Rooij, 2005; Asher and McCready, 2007; Keshet and Abney, 2024). (1) and (2) illustrate MS.

(1) [A wolf]$_i$ might come in. It$_i$ would growl.

(2) [A wolf]$_i$ might come in. #It$_i$ growls.

As illustrated in (1), an indefinite introduced within the scope of *might* brings a "hypothetical entity" into the discourse[1]. To anaphorically refer to this entity, the subsequent discourse must align with the hypothetical scenario in which the entity is assumed to exist, which is typically signaled by the use of *would* in English. The absence of *would*, as demonstrated in (2), blocks this alignment, thereby preventing the pronoun from referring to the indefinite and resulting in a failure of MS.

[1]Here, we focus on the analysis of the *de dicto* reading. While example (1) also allows a *de re* reading, where *a wolf* scopes over *might*, a detailed analysis of this reading within DTS is beyond the scope of this paper.

Proceedings of the Second Workshop on the Bridges and Gaps between Formal and Computational Linguistics (BriGap-2), pages 15–19

2 Dependent Type Semantics

DTS is a framework developed within the propositions-as-types paradigm. In DTS, the notion of existential quantification $\exists x \in A.B$ is represented by the *dependent product types* $(x : A) \times B$, which are types of pairs (a, b) such that a is of type A and b is of type $B(a)$. The SR of the unmodalized sentence in (3a) is given in (3b). We employ vertical notation for the dependent product type in the subsequent discussion, and π_1 denotes the proof constructor that yields the first projection of such a pair.

(3) a. A wolf came in.

b. $\left[\begin{array}{l} u : \left[\begin{array}{l} x : \textbf{entity} \\ \textbf{wolf}(x) \end{array}\right] \\ \textbf{comeIn}(\pi_1(u)) \end{array}\right]$

Bekki (2023) analyzes pronouns as introducing *underspecified types*, written as $(x@A) \times B$. Here, the variable x functions as a placeholder that is to be replaced by a proof of type A from a given context. Example (4) briefly illustrates how anaphora resolution proceeds in DTS.

(4) [A wolf]$_i$ came in. It$_i$ growled.

a. $\left[\begin{array}{l} v : \left[\begin{array}{l} u : \left[\begin{array}{l} x : \textbf{entity} \\ \textbf{wolf}(x) \end{array}\right] \\ \textbf{comeIn}(\pi_1(u)) \end{array}\right] \\ \left[\begin{array}{l} w@\left[\begin{array}{l} z : \textbf{entity} \\ \neg\, \textbf{human}(z) \end{array}\right] \\ \textbf{growl}(\pi_1(w)) \end{array}\right] \end{array}\right]$

b. $\left[\begin{array}{l} v : \left[\begin{array}{l} u : \left[\begin{array}{l} x : \textbf{entity} \\ \textbf{wolf}(x) \end{array}\right] \\ \textbf{comeIn}(\pi_1(u)) \end{array}\right] \\ \textbf{growl}(\pi_1\pi_1(v)) \end{array}\right]$

The underspecified type in (4a) is eliminated through *type-checking*, a process that validates whether an SR is a well-formed type under a given context. Upon encountering $w@((z : \textbf{entity}) \times \neg\textbf{human}(z))$, the type-checking algorithm attempts to find a proof of type $(z : \textbf{entity}) \times \neg\textbf{human}(z)$. In this specific case, such a proof is successfully found and substituted for the variable x. Subsequently, $\pi_1\pi_1(v)$, which corresponds to the entity x (i.e., the first element of $\pi_1(v)$), serves as the argument of the predicate **growl**, thereby resolving the pronoun *it*.

3 Modal DTS

To account for modal expressions within DTS, we propose Modal DTS, an extension grounded in Contextual Modal Type Theory (CMTT) (Nanevski et al., 2008). Modal DTS introduces two novel type constructors: $[\Psi]$ for necessity and $\langle\Psi\rangle$ for possibility, both of which are parameterized by a context Ψ. In a manner analogous to possible worlds semantics, Ψ serves as a proxy for a domain of possible worlds; accordingly, $[\Psi]$ and $\langle\Psi\rangle$ indicate that the propositions within their scope hold in all or some worlds, respectively, where Ψ is true. As an example, Figure 1 illustrates the SR of (1).

$$\left[\begin{array}{l} v\text{: } \langle\Psi\rangle \left[\begin{array}{l} u : \left[\begin{array}{l} x : \textbf{entity} \\ \textbf{wolf}(x) \end{array}\right] \\ \textbf{comeIn}(\pi_1(u)) \end{array}\right] \\ [\Psi] \left[\begin{array}{l} w@\left[\begin{array}{l} z : \textbf{entity} \\ \neg\, \textbf{human}(z) \end{array}\right] \\ \textbf{growl}(\pi_1(w)) \end{array}\right] \end{array}\right]$$

Figure 1: SR of (1) before anaphora resolution

As described in § 2, a dependent product type is a type of pairs where the second conjunct depends on the first element of the pair, i.e., the second conjunct is within the scope of the dependent product type. Accordingly, in Figure 1, where the SR of the first sentence forms the first conjunct, and that of the second sentence the second conjunct, the continuation *it would growl* quantifies over the subset of possible worlds in which *a wolf came in*.

3.1 Contextual Modal Type Theory

Intuitionistic modal logic for necessity is founded on the judgmental notion of categorical truth. Nanevski et al. (2008) examined the consequences of relativizing these notions of categorical truth to explicitly specified contexts, resulting in the formulation of contextual modal logic and its type-theoretic counterpart. Nanevski et al. (2008) advanced the structural approach to intuitionistic modal logic by allowing arbitrary contexts to be internalized within propositions. From a type-theoretic standpoint, CMTT is based on contextual modal logic and provides formal definitions for proof term assignment, substitution on terms, proof reductions and expansions, as well as strong normalization. From a logical standpoint, CMTT constitutes a relativized variant of the intuitionistic modal logic S4.

Modal DTS is a framework that uniquely integrates the dependent types of DTS with the modal types of CMTT. The newly introduced types are grounded in the notions of *contextual necessity* and *contextual possibility* as defined in CMTT. Contextual necessity, denoted as $[\Psi]A$, indicates that

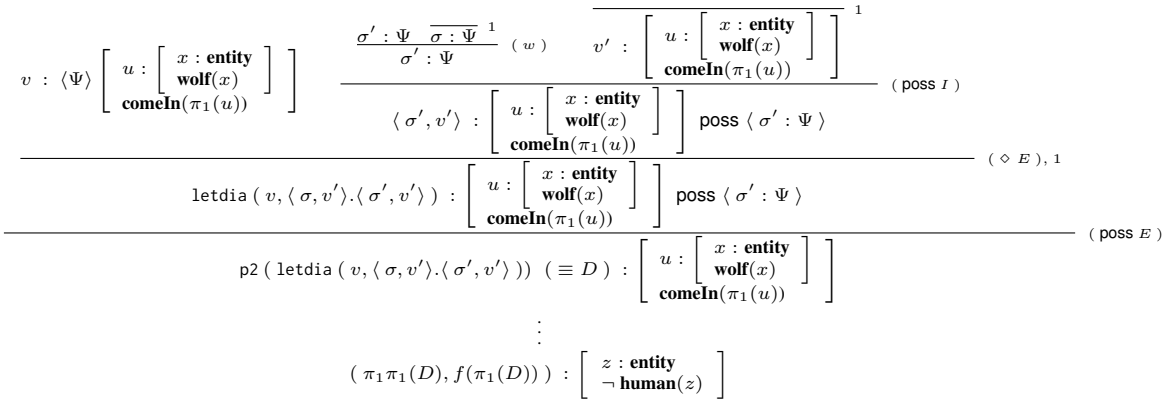


Figure 2: A part of the proof search for (1) A wolf might come in. It would growl.

A is true in all worlds in Ψ. Contextual possibility introduces a new judgment, A poss$\langle\Psi\rangle$, which expresses existential quantification over possible worlds: the judgment A poss$\langle\Psi\rangle$ holds if there exists a world in which both Ψ and A are simultaneously true. This judgment is internalized via the type operator $\langle\Psi\rangle A$.

3.2 New rules added to DTS

DTT defines types using three rules: the formation rule, the introduction rule, and the elimination rule. The formation rule specifies the conditions under which a type is well-formed; the introduction rule describes how a type can be constructed; and the elimination rule explains how a type can be used.

For the modal types $[\Psi]A$ and $\langle\Psi\rangle A$, we define their formation rules independently, while adopting their introduction and elimination rules from CMTT. In the case of A poss$\langle\Psi\rangle$, we define its formation and elimination rules independently, and adopt its introduction rule from CMTT [2].

4 Analysis

Our account treats anaphora resolution as the specification of an underspecified type by proof construction. The issue of anaphora accessibility in MS, as observed in examples (1) and (2), is thereby reduced to the question of whether the placeholder can be filled by replacing the variables of underspecified types with concrete proof terms during type-checking.

4.1 Sentence to be modally subordinated

As an example of successful proof construction, we consider (1). Figure 2 illustrates how the variable w associated with the underspecified type in Figure 1 is removed by a proof of type $(z : \textbf{entity}) \times \neg\textbf{human}(z)$. Here, `letdia` denotes the proof term for the elimination rule of $\langle\Psi\rangle$, and `p2` denotes that for the elimination rule of poss.

The strategy for constructing the proof is to first obtain D (`p2 ( letdia ( `$v, \langle\, \sigma, v'\rangle.\langle\, \sigma', v'\rangle$ `) )`) in Figure 2, which is a proof of the SR of the first sentence with the modal expression eliminated, i.e., $(u : ((x : \textbf{entity}) \times \textbf{wolf}(x))) \times \textbf{comeIn}(\pi_1(u))$. The proof search then proceeds as a standard procedure in DTT without involving modal expressions. Consequently, as illustrated by the dotted line in Figure 2, it is sufficient to find a proof of type $(z : \textbf{entity}) \times \neg\textbf{human}(z)$ using only the existing rules for dependent types.

Let us follow the proof diagram in Figure 2 step by step. Since the SR of the second sentence can depend on the proof of the first, the SR of the first sentence can be used in the proof of the second. To utilize the SR of the first sentence within the scope of possibility, we apply the elimination rule for $\langle\Psi\rangle$ (= $\Diamond E$). This application requires that the SR of the sentence be introduced within a poss environment, as derived from CMTT. Therefore, we apply the introduction rule for poss (= poss I) to introduce the poss environment, using the context $\sigma' : \Psi$ provided by the formation rule for necessity associated with the second sentence's modal expression $[\Psi]$. Finally, the elimination rule for poss (= poss E) is applied to eliminate the poss environment, and we seek the proof D of the SR of the first sentence, stripped of its modal expression.

After returning to a general proof search without modal expressions, we search for a proof of type $(z : \textbf{entity}) \times \neg\textbf{human}(z)$, using the information "if something is a wolf, then it is not human" as a

[2] For details on the new rules added to DTS, please refer to https://github.com/iimuraaoi/modal_dts.git

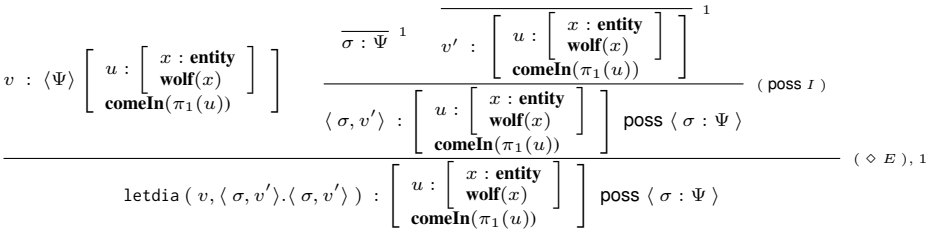


Figure 3: A part of the proof search for (2) A wolf might come in. #It growls.

global function f in the context. As a result, the variable w associated with the underspecified type in Figure 1 is removed, yielding Figure 4.

$$\left[\begin{array}{l} v\colon \langle\Psi\rangle \left[\begin{array}{l} u : \left[\begin{array}{l} x : \textbf{entity} \\ \textbf{wolf}(x) \end{array}\right] \\ \textbf{comeIn}(\pi_1(u)) \end{array}\right] \\ [\Psi]\ \ \textbf{growl}(\pi_1\pi_1(D)) \end{array}\right]$$

Figure 4: SR of (1) after anaphora resolution

4.2 Sentence not to be modally subordinated

On the other hand, the anaphora in (2) is unacceptable. Modal DTS analyzes anaphora accessibility, as DTS does, in terms of proof constructability.

$$\left[\begin{array}{l} v\colon \langle\Psi\rangle \left[\begin{array}{l} u : \left[\begin{array}{l} x : \textbf{entity} \\ \textbf{wolf}(x) \end{array}\right] \\ \textbf{comeIn}(\pi_1(u)) \end{array}\right] \\ \left[\begin{array}{l} w@ \left[\begin{array}{l} z : \textbf{entity} \\ \neg\, \textbf{human}(z) \end{array}\right] \\ \textbf{growl}(\pi_1(w)) \end{array}\right] \end{array}\right]$$

Figure 5: SR of (2) before anaphora resolution

The reason why proof construction is blocked in the anaphora resolution of (2) lies in the rule (poss E), which is defined independently in Modal DTS. The application of (poss E) imposes a restriction: the proof term corresponding to the context used to introduce the poss environment must appear as a free variable in the overall proof term prior to the application of (poss E). In Figure 2, for example, the application is permitted because this condition is satisfied.

$$\sigma' \in \texttt{letdia}\,(\, v, \langle\, \sigma, v'\rangle.\langle\, \sigma', v'\rangle\,)$$

Stated differently, this constraint effectively requires that if the antecedent sentence contains a modal expression, then the consequent sentence must also contain a modal expression. In Figure 3, which illustrates the proof search for (2), the only candidate for a modal expression introduced into the poss environment is the antecedent *might*, which fails to satisfy the condition necessary for eliminating the poss environment.

$$\sigma \notin \texttt{letdia}\,(\, v, \langle\, \sigma, v'\rangle.\langle\, \sigma, v'\rangle\,)$$

In other words, the underspecified types in Figure 5 cannot be removed during the type-checking process, which accounts for the unacceptability of the sentence.

5 Conclusion

Modal DTS is more than a computational framework for modal expressions; it also offers a theoretical contribution that puts forth empirical claims within formal linguistics, thereby bridging the computational and empirical domains of natural language semantics.

Our future work will extend Modal DTS to provide a unified account of sentences involving modal expressions such as *may* and *will*, which present distinct contexts from those of *might* and *would*. Furthermore, given a variety of analyses of modal subordination proposed in formal semantics, a next step will be to conduct empirical comparisons with these alternative accounts.

Acknowledgments

We are grateful to the two anonymous reviewers for their insightful comments. This research was partially supported by JST CREST Grant Number JPMJCR20D2, Japan, and JSPS KAKENHI Grant Number JP23H03452, Japan.

References

Lasha Abzianidze. 2015. A tableau prover for natural logic and language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2492–2502.

Nicholas Asher and Eric McCready. 2007. Were, would, might and a compositional account of counterfactuals. In *Journal of Semantics*, volume 24, pages 93–129. Oxford University Press.

Daisuke Bekki. 2023. A proof-theoretic analysis of weak crossover. In *New Frontiers in Artificial Intelligence*, pages 228–241. Springer.

Daisuke Bekki and Koji Mineshima. 2017. Context-passing and underspecification in dependent type semantics. In Stergios Chatzikyriakidis and Zhaohui Luo, editors, *Studies of Linguistics and Philosophy*, pages 11–41. Springer International Publishing.

Anette Frank and Hans Kamp. 1997. On context dependence in modal constructions. In *Proceedings of Semantics and Linguistic Theory*, volume 7, pages 151–168. Linguistic Society of America.

Stefan Kaufmann. 2000. Dynamic context management. In *Formalizing the Dynamics of Information*, pages 171–188. The University of Chicago Press.

Ezra Keshet and Steven Abney. 2024. Intensional anaphora. In *Semantics and Pragmatics*, volume 17, pages 1–54. Linguistic Society of America.

Per Martin-Löf. 1984. Intuitionistic type theory. volume 17. Bibliopolis.

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.

Aleksandar Nanevski, Frank Pfenning, and Brigitte Pientka. 2008. Contextual modal type theory. In *ACM Transactions on Computational Logic*, volume 9, pages 1–49. Association for Computing Machinery.

Craige Roberts. 1989. Modal subordination and pronominal anaphora in discourse. In *Linguistics and Philosophy*, volume 12, pages 683–721. Springer.

Ribeka Tanaka, Koji Mineshima, and Daisuke Bekki. 2015. Resolving modal anaphora in dependent type semantics. In *New Frontiers in Artificial Intelligence*, pages 83–98. Springer.

Robert van Rooij. 2005. A modal analysis of presupposition and modal subordination. In *Journal of Semantics*, volume 22, pages 281–305. Oxford University Press.

Exploring Gaps in the APS: Direct Minimal Pair Analysis in LLM Syntactic Assessments

Timothy Pistotti
University of Auckland

Jason Brown
University of Auckland

Michael Witbrock
University of Auckland

Abstract

Recent studies probing the Argument from the Poverty of the Stimulus (APS) have applied Large Language Models (LLMs) to test the learnability of complex syntax through surprisal-based metrics. However, divergent conclusions raise questions concerning the insights these metrics offer. While Wilcox et al. (2024) used direct minimal pair comparisons (the "wh-effect") to demonstrate that models successfully generalise knowledge of filler-gap dependencies, Lan et al. (2024) used a Difference-in-Differences (DiD) metric and found that models largely fail on parasitic gaps (PGs). This paper argues that the direct minimal pair approach offers greater diagnostic transparency. We demonstrate this by generating a full 8-permutation paradigm of refined PG stimuli and evaluating the GPT-2 model used in previous studies with a systematic Wilcox-style wh-effect analysis. Our results show that GPT-2 succeeds across all four tested conditions, indicating robust knowledge of filler-gap licensing principles even in complex PG environments. This finding, which contrasts with the more ambiguous results from DiD-style metrics, suggests that the choice of evaluation metric is critical for assessing an LLM's syntactic competence.

1 Introduction

The evaluation of syntactic knowledge in Large Language Models (LLMs) has become a crucial area of research for understanding their capabilities and for empirically addressing foundational questions in linguistics, such as the Argument from the Poverty of the Stimulus (APS). Surprisal, the negative log probability of a word given its context, has emerged as a key psycholinguistic metric for these evaluations (Linzen et al., 2016; Futrell et al., 2019; Wilcox et al., 2024).

Recent work has employed surprisal-based metrics to test LLM knowledge of complex dependencies, yet has adopted different evaluation paradigms. Wilcox et al. (2024) investigated various filler-gap dependencies by measuring a "wh-effect," a direct surprisal comparison between minimal pairs that differ only in the presence of a *wh*-filler versus a complementizer *that*. Their findings generally indicated that LLMs successfully acquire knowledge of these structures.

In response, Lan et al. (2024) tested more complex phenomena—parasitic gaps (PGs) and across-the-board (ATB) movement. To do so, they introduced a Difference-in-Differences (DiD) metric, a statistical tool designed to measure an interaction effect across a 2×2 paradigm of stimuli. Their findings, showing poor LLM performance on PGs and ATB movement, were interpreted as support for the APS.

While both approaches have merit, this paper argues that they differ greatly in their diagnostic transparency. The direct minimal pair approach allows for clear, interpretable tests of specific linguistic hypotheses. We apply this more direct framework to the PG phenomenon and find that the model's knowledge is more robust than suggested by prior work, indicating that the choice of metric can significantly shape conclusions about model competence.

2 Analysis of Evaluation Paradigms

Though Lan et al. (2024) and Wilcox et al. (2024) both rely on surprisal-based evaluation of LLMs on syntactic phenomena, the specific comparisons made differ in their diagnostic power. Here, we detail the distinct approaches taken by each paper, summarised in Table 1.

The method used by Wilcox et al. (2024) relies on direct minimal pair comparisons where only a single variable is manipulated while the critical region remains identical. This approach offers high interpretability, as the resulting surprisal difference (the wh-effect) can be uniquely attributed to the

Proceedings of the Second Workshop on the Bridges and Gaps between Formal and Computational Linguistics (BriGap-2), pages 20–25

Paper	Prediction	Metric / Evaluation Method
Wilcox et al. (2024)	1. Gaps require an upstream filler.	**Wh-Effect (+gap):** The surprisal at post-gap material should be lower with a *wh*-filler than with *that*. *Metric:* $S(w^+\|C_{what}) - S(w^+\|C_{that}) < 0$
	2. Fillers require a downstream gap.	**Wh-Effect (-gap):** The surprisal at an overt NP filling a potential gap site should be higher with a *wh*-filler than with *that*. *Metric:* $S(w^-\|C_{what}) - S(w^-\|C_{that}) > 0$
Lan et al. (2024)	1. An LLM should prefer the grammatical multi-gap PG structure over its ungrammatical counterpart where the main clause gap (G2) is filled.	**Direct Preference:** Compares the surprisal of the gapped vs. ungapped G2 continuation in a `+Filler` context. *Metric:* $\Delta_{+\text{filler}} > 0$, where $\Delta = S(\text{ungapped}) - S(\text{gapped})$.
	2. The model's preference for a gapped G2 should be stronger when licensed by a *wh*-filler than when it is absent.	**Difference-in-Differences (DiD):** Compares the preference for a gap (Δ) across `+Filler` and `-Filler` contexts. *Metric:* $\Delta_{+\text{filler}} > \Delta_{-\text{filler}}$

Table 1: Comparison of core predictions and evaluation metrics. Wilcox et al. (2024) focus on direct minimal pairs where only the filler is manipulated. Lan et al. (2024) use a 2×2 paradigm to calculate an overall interaction effect (DiD).

model's reaction to the manipulated variable.

In contrast, the DiD metric employed by Lan et al. (2024) is necessitated by a paradigm where the critical words being compared are not identical. Here, direct comparison is confounded by the baseline lexical probabilities of the differing critical words. To illustrate, consider the representative example (item 2 from the Lan et al. (2024) project's dataset) shown in Table 2

Condition	Critical Word	Surprisal (bits)
'+Filler, +Gap1, -Gap2'	"you"	4.14
'+Filler, +Gap1, +Gap'	"soon"	22.98
'-Filler, -Gap1, -Gap'	"you"	5.77
'-Filler, -Gap1, +Gap2'	"soon"	23.34

Table 2: Surprisal values for the critical word in each of the four conditions for "I know **who/that** Bob's talking to **(Jennifer)** is about to bother (**you**) **soon**."

Calculating their direct preference metric, $\Delta_{+\text{filler}} = S(\text{you}) - S(\text{soon})$, yields a heavily skewed value of $4.14 - 22.98 = \mathbf{-18.84}$ bits. A large negative result like this, which may well result from the much lower frequency of the word "soon" than "you" in training data, makes it impossible to interpret the simple delta as a meaningful measure of syntactic preference. This is not an isolated case; out of the 8,064 items, we find an average baseline surprisal difference of approximately 11.5 bits between the adverbial (`gap`) and nominal (`-gap`) critical words across all conditions.

The DiD metric aims to resolve this issue by measuring the interaction effect, partially controlling for this baseline difference. However, this approach obscures the specific linguistic knowledge being tested. A large DiD effect shows that the model is sensitive to the filler's role, but does not, on its own, disentangle the distinct principles of PG licensing. This is further complicated by the fact that the '-Filler' conditions also manipulate the status of the G1 gap, preventing a clean baseline.

3 Methods

To achieve a more diagnostically precise evaluation of LLM knowledge of PGs, our approach centres on direct minimal pair comparisons. This requires a full set of stimuli to test the distinct syntactic constraints that constitute knowledge of the complex domain of parasitic gaps.

3.1 Stimulus Dataset

Using Gemini 2.5 as the generative model, we created a controlled dataset of 40 items (320 sentences total), containing all 8 permutations for each PG item given the variable conditions: $\pm$filler, $\pm$gap 1, and $\pm$ gap 2. The stimuli used unambiguous subject island structures (e.g., "the story about _") and were manually vetted for pragmatic plausibility. From this set, 33 well-formed items (264 sentences) were used for analysis after excluding 7 for verb selection issues that rendered some conditions ungrammatical (see Appendix A for a sample of the resulting data).

3.2 Analytical Framework and Procedure

Our framework applied the wh-effect metric ($S(\text{+Filler}) - S(\text{-Filler})$) across the four possible gap configurations present in our 8-permutation paradigm. This resulted in four direct minimal pair tests (P1–P4), outlined in Table 3.

Test	Gap Context	Minimal Pair Comparison	Expected Outcome
P1 *Licensing*	`+G1 , +G2` *(Full PG)*	'+F, +G1, +G2' vs. '*-F, +G1, +G2' *Tests if the wh-filler licenses the full grammatical PG dependency compared to 'that'.*	$S(\text{+F}) < S(\text{-F})$
P2 *Licensing*	`-G1 , +G2` *(Simple Ext.)*	'+F, -G1, +G2' vs. '*-F, -G1, +G2' *Tests if the wh-filler licenses a simple host gap (G2) when the parasitic gap (G1) is filled.*	$S(\text{+F}) < S(\text{-F})$
P3 *Violation*	`+G1 , -G2` *(PG, No Host)*	'*+F, +G1, -G2' vs. '*-F, +G1, -G2' *Tests the effect of a wh-filler when the host gap is filled, leaving an unlicensed PG.*	(Exploratory)
P4 *Violation*	`-G1 , -G2` *(No Gaps)*	'*+F, -G1, -G2' vs. '-F, -G1, -G2' *Tests if a wh-filler creates surprisal when no gaps are available to be licensed.*	$S(\text{+F}) > S(\text{-F})$

Table 3: Proposed Wilcox-style minimal pair comparisons for parasitic gaps. Each test compares a '+Filler' sentence ('who') to a '-Filler' sentence ('that') while holding the gap configuration constant. The expected outcome refers to the surprisal at the identical critical region.

The procedure was as follows: (1) We obtained BPE-level surprisals from GPT-2 for all 264 sentences. (2) Surprisals were aggregated for predefined critical regions by summing the surprisals of their constituent BPEs. A critical region was defined as the overt NP filling a gap (for '-gap' conditions) or the material immediately following the gap (for '+gap' conditions). (3) For each hypothesis (P1–P4), we calculated the per-item surprisal difference between the two sentences in the minimal pair. (4) one-sample t-tests were used to evaluate the significance of these mean differences.

4 Results

We evaluated GPT-2 on our new dataset, using the 33 well-formed items that passed our grammaticality checks. This section first presents the results using the Δ-based metrics before applying the more diagnostically expressive minimal pair framework.

4.1 Applying Metrics from Lan et al. (2024) to the Dataset

We calculated the direct preference ($\Delta_{+\text{filler}}$) and DiD using the four paradigm conditions corresponding to the 2×2 design. The accuracy scores are presented below, and visualised in Figure 1.

- For the direct preference criterion ($\Delta_{+\text{filler}} > 0$), GPT-2 achieves an accuracy of only **51.5%**, which is at chance level. The mean effect is positive but not statistically significant (Mean = 2.17 bits, t(32) = 1.49, $p = .072$).
- For the DiD criterion ($\Delta_{+\text{filler}} > \Delta_{-\text{filler}}$), GPT-2 achieves an accuracy of **87.9%**. The mean DiD effect is large and highly significant (Mean = 5.17 bits, $t(32) = 7.11, p < .0001$).

While the highly significant DiD result might indicate that GPT-2 has acquired robust knowledge of PGs when tested on this dataset, the chance-level performance on the direct preference metric provides no real insight concerning the linguistic capabilities of the model.

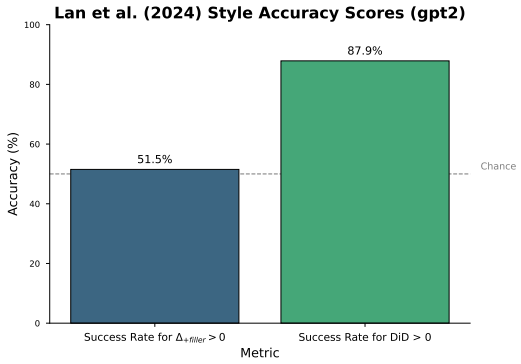


Figure 1: Results of Lan et al. metrics on our dataset

4.2 Fine-Grained Minimal Pair Analysis

We applied the Wilcox-style wh-effect analysis across the four gap configurations in our paradigm. The results, summarised in Table 4 and visualised in Figure 2, reveal a consistent pattern of success.

Hypothesis	Mean (bits)	t-statistic	p-value
P1 (+G1, +G2)	-2.61	-5.95	**< .0001**
P2 (-G1, +G2)	-3.50	-7.59	**< .0001**
P3 (+G1, -G2)	1.32	4.12	**0.0002**
P4 (-G1, -G2)	4.22	10.02	**< .0001**

Table 4: Mean Wilcox-style wh-effects ($S(\text{+F}) - S(\text{-F})$) and statistics from one-sample t-tests (N=33 items). Significant results (p < .05) are in bold.

The results show a clear pattern of success. In the two grammatical licensing contexts, **P1** (full

PG) and **P2** (simple extraction), the model correctly finds the sentences with a *wh*-filler significantly less surprising than their ungrammatical counterparts with *that*, as indicated by the large negative mean effects ($p < .0001$ for both).

Furthermore, in the two violation contexts, the model performs as expected. For **P4**, where there are no gaps to license, the model finds the sentence with a *wh*-filler significantly more surprising than the grammatical baseline with *that* ($p < .0001$). For the exploratory **P3** context, where the parasitic gap is unlicensed, the model also shows a significant positive wh-effect, robustly penalising the '+Filler' condition ($p = 0.0002$). These results indicate that GPT-2 has acquired a generalisable knowledge of filler-gap licensing that applies consistently across these complex structural variations.

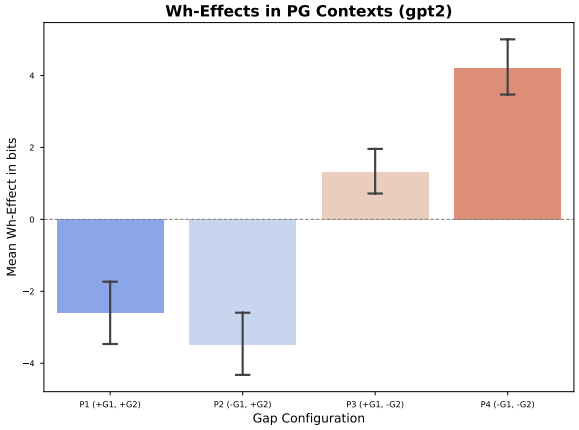


Figure 2: Mean wh-effects for the four gap configurations. Error bars represent 95% confidence intervals. All effects are in the predicted direction and statistically significant.

5 Discussion

Our fine-grained analysis, using direct minimal pair comparisons in the style of Wilcox et al. (2024), reveals a consistent and surprisingly systematic knowledge of filler-gap dependencies in GPT-2, even within the complex syntactic environment of parasitic gaps (PGs). The model correctly distinguished grammatical from ungrammatical sentences across all four of our targeted licensing and violation contexts (P1–P4), with all effects being highly statistically significant.

This finding is particularly striking when contrasted with prior work. An unexpected outcome of our study emerged when we applied the Δ-based metrics to our dataset. GPT-2's accuracy on the Difference-in-Differences (DiD) metric rose to **87.9%** from the **68.8%** reported by Lan et al. (2024) on their stimuli. Even more dramatically, the direct preference accuracy ($\Delta_{+\text{filler}} > 0$) jumped from a reported **5.6%** to **51.5%** on our dataset.

We hypothesize that this marked improvement is not necessarily because the underlying linguistic challenge was simplified, but because the stimuli themselves are more representative of canonical PGs and free from specific confounds. This suggests that surprisal-based evaluations of complex syntax are highly sensitive to stimulus quality. The original conclusion that GPT-2 fails to learn PGs may have been at least partially influenced by unintended lexical ambiguities and structural complexities in the test data, rather than solely due to failure to acquire the syntactic generalisation itself.

This highlights the primary methodological takeaway: the choice of evaluation metric profoundly impacts the conclusions drawn about a model's capabilities. While a single interaction metric like the DiD can identify a general sensitivity to a licensor, it can obscure the details of what a model has learned. Our fine-grained P1-P4 analysis, by isolating specific linguistic principles, provides a more transparent and diagnostically powerful tool for building a more accurate picture.

6 Conclusion

This paper contrasted two prominent methods for evaluating LLM syntactic knowledge and argued for the superior diagnostic clarity of a fine-grained analysis based on direct minimal pair comparisons. Our results, using a new controlled dataset, indicate that GPT-2's knowledge of the principles governing parasitic gaps is more robust than previously shown. This suggests that conclusions about model capabilities are highly sensitive to both stimulus quality and the chosen evaluation metric.

We advocate that future research adopt more direct and interpretable tests. A logical next step is to apply this framework to the other models tested by Lan et al. (2024), which performed even more poorly on the original dataset, to see whether performance there is similarly sensitive to stimulus design, or whether fine-grained analysis provides insights into what aspect of the PG licensing the model has failed to acquire. This approach promises a more rigorous foundation for claims about model capabilities and their implications for debates concerning the Argument from the Poverty of the Stimulus.

References

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. 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 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Nur Lan, Emmanuel Chemla, and Roni Katzir. 2024. Large language models and the argument from the poverty of the stimulus. *Linguistic Inquiry*, pages 1–28.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Ethan Gotlieb Wilcox, Richard Futrell, and Roger Levy. 2024. Using computational models to test syntactic learnability. *Linguistic Inquiry*, 55(4):805–848.

A Sample of Generated Stimuli

Below is a sample of our generated dataset presented in comma-separated value format. Note that Item 2, which uses the anti-rogative "believed," is included here as an example of one of the 7 item sets excluded from our final analysis. This item was excluded because the main verb does not license a *wh*-complement, rendering the '+filler' conditions ungrammatical and thus unsuitable for the intended minimal pair comparisons.

```
sentence_type,item_id,condition,full_sentence
subject_pg_full,1,plusF_plusG1_plusG2,The investigators know who the story about is
    likely to damage severely.
subject_pg_full,1,plusF_plusG1_minusG2,The investigators know who the story about is
     likely to damage the campaign severely.
subject_pg_full,1,plusF_minusG1_plusG2,The investigators know who the story about
    the politician is likely to damage severely.
subject_pg_full,1,plusF_minusG1_minusG2,The investigators know who the story about
    the politician is likely to damage the campaign severely.
subject_pg_full,1,minusF_plusG1_plusG2,The investigators know that the story about
    is likely to damage severely.
subject_pg_full,1,minusF_plusG1_minusG2,The investigators know that the story about
    is likely to damage the campaign severely.
subject_pg_full,1,minusF_minusG1_plusG2,The investigators know that the story about
    the politician is likely to damage severely.
subject_pg_full,1,minusF_minusG1_minusG2,The investigators know that the story about
     the politician is likely to damage the campaign severely.
subject_pg_full,2,plusF_plusG1_plusG2,The audience believed who the picture of might
     have flattered greatly.
subject_pg_full,2,plusF_plusG1_minusG2,The audience believed who the picture of
    might have flattered the director greatly.
subject_pg_full,2,plusF_minusG1_plusG2,The audience believed who the picture of the
    actor might have flattered greatly.
subject_pg_full,2,plusF_minusG1_minusG2,The audience believed who the picture of the
     actor might have flattered the director greatly.
subject_pg_full,2,minusF_plusG1_plusG2,The audience believed that the picture of
    might have flattered greatly.
subject_pg_full,2,minusF_plusG1_minusG2,The audience believed that the picture of
    might have flattered the director greatly.
subject_pg_full,2,minusF_minusG1_plusG2,The audience believed that the picture of
    the actor might have flattered greatly.
subject_pg_full,2,minusF_minusG1_minusG2,The audience believed that the picture of
    the actor might have flattered the director greatly.
subject_pg_full,3,plusF_plusG1_plusG2,The board understood who the critique of would
     probably anger immensely.
subject_pg_full,3,plusF_plusG1_minusG2,The board understood who the critique of
    would probably anger the CEO immensely.
subject_pg_full,3,plusF_minusG1_plusG2,The board understood who the critique of the
    new project would probably anger immensely.
subject_pg_full,3,plusF_minusG1_minusG2,The board understood who the critique of the
     new project would probably anger the CEO immensely.
subject_pg_full,3,minusF_plusG1_plusG2,The board understood that the critique of
    would probably anger immensely.
subject_pg_full,3,minusF_plusG1_minusG2,The board understood that the critique of
    would probably anger the CEO immensely.
subject_pg_full,3,minusF_minusG1_plusG2,The board understood that the critique of
    the new project would probably anger immensely.
subject_pg_full,3,minusF_minusG1_minusG2,The board understood that the critique of
    the new project would probably anger the CEO immensely.
```

Listing 1: Sample of Gemini 2.5 Generated Data

Coordination of Theoretical and Computational Linguistics

Adam Przepiórkowski
ICS Polish Academy of Sciences
and University of Warsaw
adamp@ipipan.waw.pl

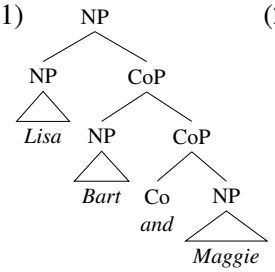
Agnieszka Patejuk
ICS Polish Academy of Sciences
aep@ipipan.waw.pl

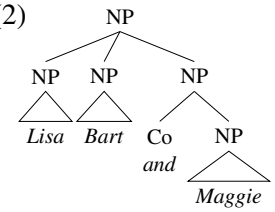
Abstract

The aim of this paper is to present a case study of a fruitful and, hopefully, inspiring interaction between formal and computational linguistics. A variety of NLP tools and resources have been used in linguistic investigations of the symmetry of coordination, leading to novel theoretical arguments. The converse impact of theoretical results on NLP work has been successful only in some cases.

1 Introduction

Coordination, as in *Lisa, Bart, and Maggie*, is a controversial theoretical linguistic phenomenon, with no agreement on its structure and properties. The two most common structures assumed in constituency syntax are those in (1)–(2), with variants of the binary structure in (1) almost universally assumed in Chomskyan linguistics, and variants of the flat structure in (2) universally adopted in LFG (Bresnan 1982, Dalrymple et al. 2019; see Patejuk 2023) and HPSG (Pollard and Sag 1987, 1994; see Abeillé and Chaves 2024).

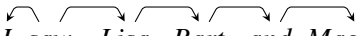
(1)


(2)


Similar disagreement is observed in theoretical dependency linguistics, see (4)–(6), and – consequently – in dependency corpora, where the current annotation standard, Universal Dependencies (UD), assumes (3), with Enhanced Dependencies (Schuster and Manning 2016) adding elements of (6), Surface-syntactic Universal Dependencies (Gerdes et al. 2018, 2021) adopts a variant of (4), and original Prague Dependency Treebanks (Hajič et al. 2006) assume (5).

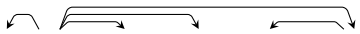
(3) Bouquet/Stanford (de Marneffe et al. 2021):

(4) Chain/Moscow (Mel'čuk 1988, 2009):



(5) &-headed/Prague (Sgall et al. 1986):

(6) Multi-headed/London (Hudson 1984, 1990):



This variety of structures reflects the lack of agreement regarding the fundamental issue of the symmetry of coordination: are all conjuncts syntactically equal? On some approaches, e.g., (1) and (3)–(4), the first conjunct is (closest to) the head of the coordinate structure and so it largely determines the properties of coordination. On other approaches, e.g., (2) and (5)–(6), all conjuncts determine such properties to the same extent.

This issue is related to another bone of contention: do conjuncts have to be alike, or can unlike categories or different grammatical functions be coordinated? Assuming that unlike category coordination is possible, as in the attested (7) (from the English Web 2015 corpus[1]), asymmetric approaches predict that the whole coordinate structure is an NP, while on symmetric approaches it has features of both NP and CP; see Figure 1.

(7) *I understand* [[$_{\text{NP}}$ *those concerns*] *and* [$_{\text{CP}}$ *that they are sincerely held*]].

For decades, these issues have been discussed on the basis of a handful of – usually constructed –

[1] http://www.sketchengine.eu (Jakubíček et al. 2013)

Proceedings of the Second Workshop on the Bridges and Gaps between Formal and Computational Linguistics (BriGap-2), pages 26–34

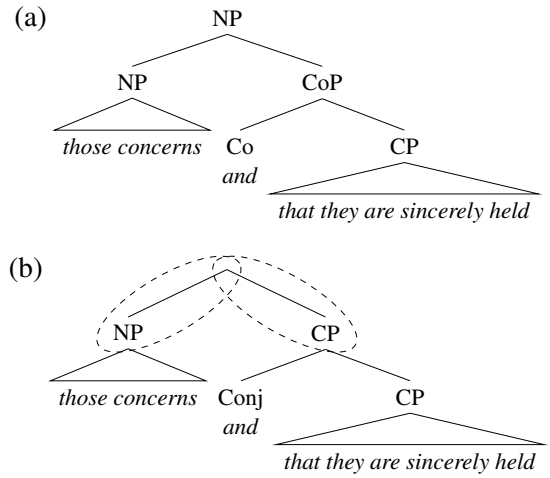


Figure 1: The structure of the coordination in (7) on the asymmetric approach of Munn 1993 (in (a)) and on the symmetric approach of Neeleman et al. 2023 (in (b))

examples and have remained unresolved. This paper shows that employing computational linguistic tools and resources makes it possible to construct novel theoretical arguments and that, conversely, awareness of theoretical issues may sometimes influence such tools and resources.

2 Morphosyntactic Corpora

The most basic use of NLP technologies that a theoretical linguist can make is the use of annotated corpora, often created with NLP applications in mind and/or annotated with NLP tools. There is surprisingly little awareness of morphosyntactically annotated corpora among generative syntacticians and of the power that lies in their associated query languages. Learning a given tagset and a given query language takes time, and tagsets and query languages may differ considerably even for a single language,[2] but doing so is well worth the effort.

For example, for many decades generative linguists believed, following a remark in Chomsky 1957, that only the same syntactic categories can be coordinated, a belief that was elevated to the status of a universal law (Williams 1981) and defended against some constructed counterexamples (e.g., in Sag et al. 1985) as recently as in 2020 (Bruening and Al Khalaf 2020). In particular, Sag et al.'s (1985) examples were claimed to involve the coordination of "supercategories" Predicate (in (8)) and Modifiers (in (9)), with no similar coordination of different categories possible in true argument positions.

(8) *Pat is* [[$_{\text{NP}}$ *a Republican*] *and* [$_{\text{AP}}$ *proud of it*]].

(9) *We walked* [[$_{\text{ADVP}}$ *slowly*] *and* [$_{\text{PP}}$ *with great care*]].

This long-held myth was refuted in Patejuk and Przepiórkowski 2023 – a paper in a prominent generative journal – on the basis of a few dozen attested examples; the effectiveness of this rebuttal was admitted by an erstwhile advocate of the refuted view, Bruening (2025): "[Patejuk and Przepiórkowski (2023)] are correct, and there is no requirement that conjuncts match in syntactic category".[3]

As coordination of unlikes is textually very rare, this would not be possible without an advanced use of morphosyntactically annotated corpora. In this case, the English Web 2015 corpus was queried with queries of varying complexity, e.g., (10) used to find examples such as (11), or (12) to find examples such as (13). Note that such queries require not only the basic knowledge of the query language and the relatively standard Penn Treebank (PTB; Marcus et al. 1993) tagset, but also the knowledge of regular expressions, not universally mastered by theoretical linguists.[4]

(10)
```
[lemma="with"] [lemma="respect|dignity"]
[tag="CC"] [tag="RB"]
```

(11) *. . . not all of us treat our animals* [[$_{\text{PP}}$ *with respect*] *and* [$_{\text{ADVP}}$ *humanly*]]*!*

(12)
```
[lemma="teach" & tag="VV.*"]
[tag="N.*|P.*|JJ.*|DT|CD.*"]{1,5} "that"
[tag="N.*|P.*|JJ.*|DT|CD.*"] []{1,5}
[word=","]? [tag="CC"] [tag="TO"]
```

(13) *You teach me* [[$_{\text{CP}}$ *that hard work pays off*] *and* [$_{\text{INFP}}$ *to never give up on a goal*]].

Among the claims of likeness of conjuncts is the claim that, in languages with rich nominal morphology, only the same grammatical cases can be coordinated (Weisser 2020). Advanced queries applied to the National Corpus of Polish (Przepiórkowski et al. 2011, 2012) and the Turkish Web 2012 corpus (Baisa and Suchomel 2012) helped to show that both kinds of coordination of unlikes – unlike categories and unlike cases – are readily found

[2] For example, those of The Corpus of Contemporary American English (COCA; Davies 2008–2025) are very different from those of enTenTen English corpora made available via SketchEngine (see fn. 1).

[3] See also Przepiórkowski and Patejuk 2025.

[4] While "`[tag="N.*|P.*|JJ.*|DT|CD.*"]{1,5}`" in (12) is a very poor regular definition of a nominal phrase, it returns a reasonable number of true positives.

in Polish (Przepiórkowski 2022) and in Turkish (Şenşekerci and Przepiórkowski 2024). Again, the intimate knowledge of relevant tagsets and query languages was crucial – for example, the awareness of the special feature of the Poliqarp search engine (Janus and Przepiórkowski 2007) of the Polish corpus, which makes it possible to use variables to specify that a given token must have a different case value than some other token.[5]

3 Valency Dictionaries

A resource that could be useful in an investigation of unlike category coordination is a valency dictionary, i.e., a lexicon containing information about arguments of verbs (and possibly predicates of other categories). Such a lexicon could encode information that a given argument of a given verb, e.g., the object of *understand* (cf. (7) above), could be realized as, say, an NP or a CP, which would make it worth checking whether this argument can be realized as NP and CP coordinated. Unfortunately, neither traditional valency dictionaries, nor machine-readable lexicons such as VerbNet (Kipper et al. 2006), contain such information.

Fortunately, however, the development of the largest and most detailed Polish valency dictionary, *Walenty* (Przepiórkowski et al. 2014, 2017), was informed by this issue, which resulted in the following unique feature. In this human- and machine-readable lexicon, arguments are described as sets of categories, often singleton sets, signaling that a given argument must bear a specific category, e.g., an NP. However, when there is corpus evidence that a given argument may be realized by a number of categories and, importantly, by coordinations of such unlike categories, this argument is described as a set of these categories.

For example, a valency schema for *zapowiedzieć* 'announce' contains information about the following two arguments (where ... indicates that more elements of the set are specified in the dictionary but omitted here for clarity), among others:

(14) a. `subj{np(str)}`
b. `obj{np(str);cp(int);cp(że);...}`

(14a) specifies one argument as a structural (i.e., normally nominative) NP subject, while (14b) specifies another argument as an object which may be realized as a structural (normally accusative) NP, an interrogative CP, a CP with the complementizer *że* 'that', etc. This information is supported with example (15) from the National Corpus of Polish.

(15) *Pan prezydent zapowiedział* [[$_{NP}$ *swój* Mr. president announced self's *patronat...*] *oraz* [$_{CP}$ *że takiej ustawy na* patronage and that such bill for *pewno nie podpisze*]].
certain not sign
'The president announced [[$_{NP}$ his patronage...] and [$_{CP}$ that he will definitely not sign such a bill]].'

The developed valency dictionary has been used in subsequent theoretical publications (e.g., Przepiórkowski 2022), as a rich source of examples of coordination of unlikes in Polish.

4 Implemented Grammars

As this valency dictionary is used by a number of grammar-based syntactic parsers of Polish (Patejuk 2015, Woliński 2015), these tools are able to parse sentences containing unlike category coordination, for example, the parse in Figure 2 produced by the *Świgra 2* parser (Woliński 2019).

Moreover, one of these parsers, the LFG parser *POLFIE* (Patejuk and Przepiórkowski 2012c,a, 2017b), has in turn been extensively used to verify a number of theoretical linguistic proposals, including analyses of agreement, predication, negation, numeral phrases, so-called reflexive markers, coordination of different grammatical functions, gapping, etc. (Patejuk and Przepiórkowski 2012b, 2014, 2015, 2017a, 2018, Przepiórkowski and Patejuk 2012, 2015, 2023).

Hence, in cases described in §§3–4, theoretical considerations fruitfully influenced the development of NLP resources (valency dictionary) and tools (syntactic parsers), which in turn helped in exemplifying and verifying theoretical analyses.

5 Syntactic Corpora

The possibility of theoretical influence on NLP tools and resources depends crucially on the stage of development of these tools and resources. For example, the representation of coordination in UD has various problems, e.g., it does not distinguish flat coordinations from certain nested coordinations. This problem was discussed – and solutions were proposed – in Przepiórkowski and Patejuk 2019,

[5] https://nkjp.pl/poliqarp/help/ense3.html/#x4-90003.4

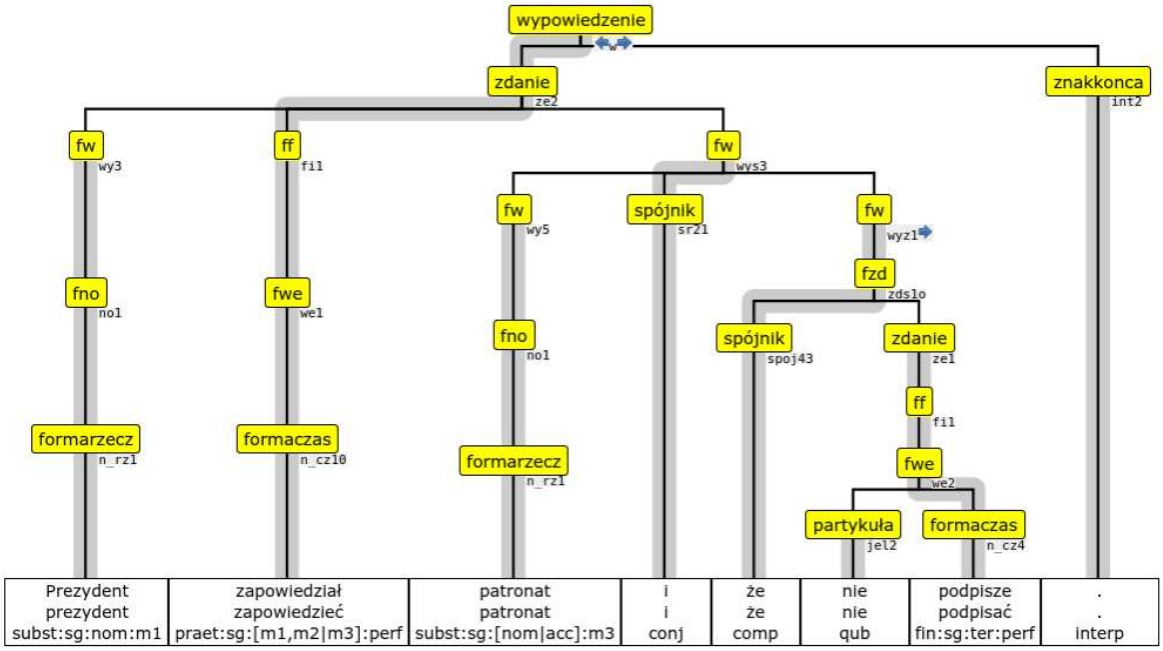


Figure 2: Parse of a much simplified version of example (15) produced by *Świgra 2*

but they have not been adopted by the UD community, probably because at this stage UD had already been employed in a number of corpora and was perceived as stable. Also other proposals, stemming from considerations of head-final languages (Kanayama et al. 2018), from the coordination of unlike grammatical functions (Patejuk and Przepiórkowski 2019), as in [*What and when*] *to eat to stay healthy*, and from the incompatibility of coordination with UD's core/non-core distinction (Przepiórkowski and Patejuk 2018), have not been adopted in UD.

Nevertheless, syntactic corpora – including UD corpora – were the basis for a novel theoretical argument against asymmetric approaches to coordination, described below.

6 Trained Parsers

Dependency Length Minimization (DLM) is a robustly demonstrated tendency for natural languages to strive for maximally local – shortest possible – dependencies.[6] Przepiórkowski and Woźniak 2023 argue that, given DLM, the distribution of lengths in binary coordinations in $\text{PTB}_{\&}$ – a version of PTB with enhanced annotation of coordination (Ficler and Goldberg 2016) – is compatible with symmetric approaches to coordination, but not with asymmetric approaches.

Specifically, they show that when the length difference between the two conjuncts increases, the tendency for the shorter constituent to be the initial conjunct also increases, but only when the governor is on the left (as in *I saw Bart and Lisa*) or absent (as in *Bart came and Lisa left*), and not when the governor is on the right (as in *Bart and Lisa laughed*). This tendency was observed whether the length of conjuncts was measured in the number of words, syllables, or characters; see Figure 3 in Appendix A. On various assumptions about the exact nature of DLM this observation is compatible with symmetric dependency representations such as (5)–(6) (and, by extension to constituency representations, (2)), but on no reasonable assumptions about DLM is it compatible with asymmetric representations such as (3)–(4) (and (1)).

This argument was reproduced on the basis of a variety of other manually annotated corpora for a number of languages, including UD corpora (Przepiórkowski et al. 2024b). However, in each case, the sparseness of data[7] resulted in some tendencies being only very weakly statistically significant and/or in the need to aggregate results in a way that might have influenced the final results.

To ameliorate this problem, a large portions of the COCA corpus of American English Davies 2008–2025 were automatically parsed with the

[6] See, e.g., Temperley and Gildea 2018 and references there, as well as Futrell et al. 2020.

[7] The relevant corpora had roughly between 15K (K = thousand) and 2,250K sentences, which translates into between 5K and 90K extracted coordinations.

Stanza dependency parser (Qi et al. 2020) trained on UD and SUD corpora, as well as with the Berkeley Neural Parser (BNP; Kitaev and Klein 2018, Kitaev et al. 2019) with the `benepar_en3` constituency model (Przepiórkowski et al. 2024a,b).[8] As these parsers produced much noise, implementing various filters removing obviously wrong parses was necessary. Nevertheless, the final results confirmed the earlier results based on manually parsed corpora, and – as was expected – all results turned out to be highly statistically significant.

While these relatively recent results have so far been only published in the proceedings of conferences devoted to NLP tools and resources (ACL, LREC-COLING, TLT), they are of vital importance for theoretical analyses of coordination, as they provide a novel argument not only against the most common treatment of coordination in corpora (both UD and SUD) and in a relatively niche theoretical dependency framework (Mel'čuk 1988, 2009), but also – by extension – against the asymmetric approaches almost universally assumed in Chomskyan linguistics.

7 Conclusion

This paper demonstrates that bridging the gap between theoretical and computational linguistics can be fruitful for both, but especially for theoretical linguistics. The awareness of traditional NLP resources (corpora, dictionaries) and tools (especially, parsers) among theoretical linguistics is too little, given how useful they can be for constructing and verifying theoretical arguments.

We conclude by noting that the NLP tools and resources used in the investigation of coordination described in this paper all date from the pre-LLM era. This is a feature, not a bug. It is not clear to us how LLMs could similarly support theoretical linguists in theory development.

[8] The relevant portions had roughly between 20M (M = million) and 70M sentences, which translates into 10–15M extracted binary coordinations.

References

Anne Abeillé and Rui Chaves. 2024. Coordination. In Stefan Müller, Anne Abeillé, Robert D. Borsley, and Jean-Pierre Koenig, editors, *Head-Driven Phrase Structure Grammar: The Handbook*, 2nd edition, pages 775–829. Language Science Press, Berlin.

Vit Baisa and Vit Suchomel. 2012. Large corpora for Turkic languages and unsupervised morphological analysis. In *First Workshop on Language Resources and Technologies for Turkic Languages, LREC 2012*, pages 28–32, Istanbul, Turkey. European Language Resources Association (ELRA).

Joan Bresnan, editor. 1982. *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, MA.

Benjamin Bruening. 2025. Selectional violations in coordination (a response to Patejuk and Przepiórkowski 2023). *Linguistic Inquiry*, 56(3):439–483.

Benjamin Bruening and Eman Al Khalaf. 2020. Category mismatches in coordination revisited. *Linguistic Inquiry*, 51(1):1–36.

Miriam Butt and Tracy Holloway King, editors. 2012. *The Proceedings of the LFG'12 Conference*. CSLI Publications, Stanford, CA.

Miriam Butt and Tracy Holloway King, editors. 2015. *The Proceedings of the LFG'15 Conference*. CSLI Publications, Stanford, CA.

Noam Chomsky. 1957. *Syntactic Structures*. Mouton, The Hague.

Mary Dalrymple, John J. Lowe, and Louise Mycock. 2019. *The Oxford Reference Guide to Lexical Functional Grammar*. Oxford University Press, Oxford.

Mark Davies. 2008–2025. The Corpus of Contemporary American English (COCA). Available online at https://www.english-corpora.org/coca/.

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

Jessica Ficler and Yoav Goldberg. 2016. Coordination annotation extension in the Penn Tree Bank. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 834–842, Berlin, Germany.

Richard Futrell, Roger P. Levy, and Edward Gibson. 2020. Dependency locality as an explanatory principle for word order. *Language*, 96(2):371–412.

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2018. SUD or Surface-Syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD. In (Marneffe et al. 2018), pages 66–74.

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2021. Starting a new treebank? Go SUD! In *Proceedings of the Sixth International Conference on Dependency Linguistics (DepLing, Syntax Fest 2021)*, pages 35–46, Sofia, Bulgaria.

Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, Zdeněk Žabokrtský, Magda Ševčíková Razímová, and Zdeňka Urešová. 2006. Prague Dependency Treebank 2.0 (PDT 2.0).

Richard Hudson. 1984. *Word Grammar*. Blackwell, Oxford.

Richard Hudson. 1990. *English Word Grammar*. Blackwell, Oxford.

Miloš Jakubíček, Adam Kilgarriff, Vojtěch Kovář, Pavel Rychlý, and Vít Suchomel. 2013. The TenTen corpus family. In *7th International Corpus Linguistics Conference CL*, pages 125–127.

Daniel Janus and Adam Przepiórkowski. 2007. Poliqarp: An open source corpus indexer and search engine with syntactic extensions. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, pages 85–88, Prague.

Hiroshi Kanayama, Na-Rae Han, Masayuki Asahara, Jena D. Hwang, Yusuke Miyao, Jinho D. Choi, and Yuji Matsumoto. 2018. Coordinate structures in Universal Dependencies for head-final languages. In (Marneffe et al. 2018), pages 75–84.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2006. Extending verbnet with novel verb classes. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC 2006*, pages 1027–1032, Genoa. ELRA.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.

Marie-Catherine de Marneffe, Teresa Lynn, and Sebastian Schuster, editors. 2018. *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*. Association for Computational Linguistics.

Igor Mel'čuk. 1988. *Dependency Syntax: Theory and Practice*. The SUNY Press, Albany, NY.

Igor Mel'čuk. 2009. Dependency in natural language. In Alain Polguère and Igor Mel'čuk, editors, *Dependency in Linguistic Description*, pages 1–110. John Benjamins, Amsterdam.

Alan Boag Munn. 1993. *Topics in the Syntax and Semantics of Coordinate Structures*. Ph.D. dissertation, University of Maryland.

Ad Neeleman, Joy Philip, Misako Tanaka, and Hans van de Koot. 2023. Subordination and binary branching. *Syntax*, 26(1):41–84.

Agnieszka Patejuk. 2015. *Unlike Coordination in Polish: An LFG Account*. Ph.D. dissertation, Institute of Polish Language, Polish Academy of Sciences, Cracow.

Agnieszka Patejuk. 2023. Coordination. In Mary Dalrymple, editor, *Handbook of Lexical Functional Grammar*, pages 309–374. Language Science Press, Berlin.

Agnieszka Patejuk and Adam Przepiórkowski. 2012a. A comprehensive analysis of constituent coordination for grammar engineering. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 2191–2207, Mumbai, India.

Agnieszka Patejuk and Adam Przepiórkowski. 2012b. Lexico-semantic coordination in Polish. In (Butt and King 2012), pages 461–478.

Agnieszka Patejuk and Adam Przepiórkowski. 2012c. Towards an LFG parser for Polish: An exercise in parasitic grammar development. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012*, pages 3849–3852, Istanbul, Turkey. European Language Resources Association (ELRA).

Agnieszka Patejuk and Adam Przepiórkowski. 2014. Structural case assignment to objects in Polish. In *The Proceedings of the LFG'14 Conference*, pages 429–447, Stanford, CA. CSLI Publications.

Agnieszka Patejuk and Adam Przepiórkowski. 2015. An LFG analysis of the so-called reflexive marker in Polish. In (Butt and King 2015), pages 270–288.

Agnieszka Patejuk and Adam Przepiórkowski. 2017a. Filling the gap. In *The Proceedings of the LFG'17 Conference*, pages 327–347, Stanford, CA. CSLI Publications.

Agnieszka Patejuk and Adam Przepiórkowski. 2017b. POLFIE: współczesna gramatyka formalna języka polskiego. *Język Polski*, XCVII(1):48–64.

Agnieszka Patejuk and Adam Przepiórkowski. 2018. Predicative constructions with infinitival and clausal subjects. In *The Proceedings of the LFG'18 Conference*, pages 304–324, Stanford, CA. CSLI Publications.

Agnieszka Patejuk and Adam Przepiórkowski. 2019. Coordination of unlike grammatical functions. In *Proceedings of the Fifth International Conference on Dependency Linguistics (DepLing, SyntaxFest 2019)*, pages 26–37. Association for Computational Linguistics.

Agnieszka Patejuk and Adam Przepiórkowski. 2023. Category mismatches in coordination vindicated. *Linguistic Inquiry*, 54(2):326–349.

Carl Pollard and Ivan A. Sag. 1987. *Information-Based Syntax and Semantics, Volume 1: Fundamentals*. CSLI Publications, Stanford, CA.

Carl Pollard and Ivan A. Sag. 1994. *Head-driven Phrase Structure Grammar*. Chicago University Press / CSLI Publications, Chicago, IL.

Adam Przepiórkowski. 2022. Coordination of unlike grammatical cases (and unlike categories). *Language*, 98(3):592–634.

Adam Przepiórkowski, Mirosław Bańko, Rafał L. Górski, and Barbara Lewandowska-Tomaszczyk, editors. 2012. *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN, Warsaw.

Adam Przepiórkowski, Mirosław Bańko, Rafał L. Górski, Barbara Lewandowska-Tomaszczyk, Marek Łaziński, and Piotr Pęzik. 2011. National Corpus of Polish. In *Proceedings of the 5th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, pages 259–263, Poznań, Poland.

Adam Przepiórkowski, Magdalena Borysiak, and Adam Głowacki. 2024a. An argument for symmetric coordination from Dependency Length Minimization: A replication study. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1021–1033, Torino, Italy. ELRA and ICCL.

Adam Przepiórkowski, Magdalena Borysiak, Adam Okrasiński, Bartosz Pobożniak, Wojciech Stempniak, Kamil Tomaszek, and Adam Głowacki. 2024b. Symmetric dependency structure of coordination: Crosslinguistic arguments from dependency length minimization. In *Proceedings of the 22nd Workshop on Treebanks and Linguistic Theories (TLT 2024)*, pages 11–22, Hamburg,Germany. Association for Computational Linguistics.

Adam Przepiórkowski, Elżbieta Hajnicz, Anna Andrzejczuk, Agnieszka Patejuk, and Marcin Woliński. 2017. Walenty: gruntowny składniowo-semantyczny słownik walencyjny języka polskiego. *Język Polski*, XCVII(1):30–47.

Adam Przepiórkowski, Elżbieta Hajnicz, Agnieszka Patejuk, Marcin Woliński, Filip Skwarski, and Marek Świdziński. 2014. Walenty: Towards a comprehensive valence dictionary of Polish. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014*, pages 2785–2792, Reykjavík, Iceland. European Language Resources Association (ELRA).

Adam Przepiórkowski and Agnieszka Patejuk. 2012. The puzzle of case agreement between numeral phrases and predicative adjectives in Polish. In (Butt and King 2012), pages 490–502.

Adam Przepiórkowski and Agnieszka Patejuk. 2015. Two representations of negation in LFG: Evidence from Polish. In (Butt and King 2015), pages 322–336.

Adam Przepiórkowski and Agnieszka Patejuk. 2018. Arguments and adjuncts in Universal Dependencies. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, pages 3837–3852, Santa Fe, NM. (Best position paper at COLING 2018).

Adam Przepiórkowski and Agnieszka Patejuk. 2019. Nested coordination in Universal Dependencies. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 58–69. Association for Computational Linguistics.

Adam Przepiórkowski and Agnieszka Patejuk. 2023. Filling gaps with Glue. In *The Proceedings of the LFG'23 Conference*, pages 223–240. PubliKon.

Adam Przepiórkowski and Agnieszka Patejuk. 2025. Prenominal adverbs, or apparent selectional violations in coordination. *Linguistic Inquiry*, Early Access:1–29.

Adam Przepiórkowski and Michał Woźniak. 2023. Conjunct lengths in English, Dependency Length Minimization, and dependency structure of coordination. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15494–15512, Toronto, Canada. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108.

Ivan A. Sag, Gerald Gazdar, Thomas Wasow, and Steven Weisler. 1985. Coordination and how to distinguish categories. *Natural Language and Linguistic Theory*, 3(2):117–171.

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced English Universal Dependencies: An improved representation for natural language understanding tasks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation, LREC 2016*, pages 2371–2378, Portorož, Slovenia. European Language Resources Association (ELRA).

Berke Şenşekerci and Adam Przepiórkowski. 2024. Coordination of unlikes in Turkish. In *The Proceedings of the LFG'24 Conference*, pages 207–225. PubliKon.

Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. Reidel, Dordrecht.

David Temperley and Daniel Gildea. 2018. Minimizing syntactic dependency lengths: Typological/cognitive universal? *Annual Review of Linguistics*, 4:67–80.

Philipp Weisser. 2020. On the symmetry of case in conjunction. *Syntax*, 23(1):42–77.

Edwin Williams. 1981. Transformationless grammar. *Linguistic Inquiry*, 12(4):645–653.

Marcin Woliński. 2015. Deploying the new valency dictionary Walenty in a DCG parser of Polish. In *Proceedings of the Fourteenth International Workshop on Treebanks and Linguistic Theories (TLT 14)*, pages 221–229, Warsaw. Institute of Computer Science, Polish Academy of Sciences.

Marcin Woliński. 2019. *Automatyczna analiza składnikowa języka polskiego*. Wydawnictwa Uniwersytetu Warszawskiego, Warsaw.

Appendix A. Results of Przepiórkowski and Woźniak 2023

(See next page.)

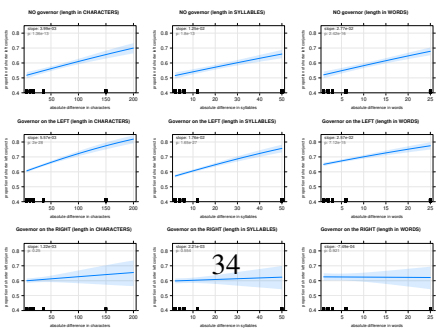


Figure 3: Modelled proportions of coordinations in PTB_{fc} with left conjuncts shorter, depending on the absolute difference of conjunct lengths, with confidence bands (Przepiórkowski and Woźniak 2023).

An instructive implementation of semantic parsing and reasoning using Lexical Functional Grammar

Mark-Matthias Zymla, Kascha Kruschwitz, Paul Zodl
Department of Linguistics
University of Konstanz
Konstanz, Germany
{mark-matthias.zymla | kascha.kruschwitz | paul.zodl}@uni-konstanz.de

Abstract

This paper presents a computational resource for exploring semantic parsing and reasoning through a strictly formal lense. Inspired by the framework of Lexical Functional Grammar, our system allows for modular exploration of different aspects of semantic parsing. It consists of a hand-coded formal grammar combining syntactic and semantic annotations, producing basic semantic representations. The system provides the option to extend these basic semantics via rewrite rules in a principled fashion to explore more complex reasoning. The result is a layered system enabling an incremental approach to semantic parsing. We illustrate this approach with examples from the Fracas testsuite, demonstrating its overall functionality and viability.

1 Introduction

Formal approaches to computational linguistics have been surpassed by quantitative methods in the fast-paced task-driven field of NLP. However, modern NLP approaches trade explainability and interpretability for performance gains. This puts a larger burden on researchers who need to evaluate whether a system captures the expected linguistic generalizations, and limits the possibility to test the effect of small tweaks to a system. Thus, understanding and exploring patterns in syntax and semantics is challenging and potentially affected by confounding factors (e.g., McCoy et al. 2019).

Formal approaches inherently require accurate descriptions of patterns. Computational approaches, in particular, often highlight wanted and unwanted interactions between linguistic descriptions. Detecting these is an essential skill of (computational) linguists and, thus, we deem it a worthwhile goal to make corresponding resources accessible. Concretely, we present a system for semantic parsing and reasoning based on Lexical Functional Grammar (LFG; Kaplan and Bresnan 1982).[1] LFG is characterized by modular but interconnected linguistic descriptions, allowing researchers to make comparatively simple statements about particular domains of language. We build on XLE+Glue (Dalrymple et al., 2020) designed for exploring the syntax/semantics interface in LFG. However, while XLE+Glue has been used in formal semantic research to verify analyses, it has not been used in a task oriented setting as is typical in computational linguistics and NLP. This is the main drawback that this paper attempts to address by extending the XLE+Glue[2] pipeline to also incorporate reasoning tools, particularly, the Vampire theorem prover (Kovács and Voronkov, 2013). This allows us to lay the foundation for task-oriented semantic parsing (i.e., for NLI). We take inspiration from the seminal work on semantic parsing by Blackburn and Bos (2005) but also consider more recent proposals, particularly Haruta et al. (2020, 2022).

The contribution of this paper is a comprehensive implementation of semantic parsing that is grounded in a rigorous formal framework. The system is designed to be accessible and extensible, building on the discipline of grammar engineering. It enforces incremental development of linguistic analyses and enables testing the interplay of these analyses in a task-oriented fashion. The paper is structured as follows: section 2 presents LFG as the formal foundation of our system. Section 3 describes the full system, focusing on the novel interface between XLE+Glue and Vampire. Section 4 presents a qualitative evaluation of the system based on examples from the Fracas testsuite. Section 5 discusses some limitations and, finally, section 6 concludes.

[1] For recent introductions see Börjars (2020); Asudeh (2022), or Dalrymple (2023).

[2] https://github.com/Mmaz1988/xleplusglue/tree/2024_inference

Proceedings of the Second Workshop on the Bridges and Gaps between Formal and Computational Linguistics (BriGap-2), pages 35–51

2 Formal background

Lexical Functional Grammar (LFG) is a grammar formalism that is well-known for its formal rigor, allowing for a faithful computational implementation in the form of the Xerox Linguistics Environment (XLE; Crouch et al. 2017). From its beginnings, it has established itself as a crosslinguistically viable tool for describing language. This is particularly highlighted by the ParGram project (Butt et al., 2002) which produced computational LFG grammars illustrated by virtue of the ParGramBank (Sulger et al., 2013).[3]

2.1 Lexical Functional Grammar

The main appeal of LFG lies in its modular architecture that allows researchers and grammar engineers to make comparatively simple statements about linguistic facts in one domain of analysis (e.g., syntax), while maintaining a clearly defined mapping to other aspects of grammar (e.g., prosody, semantics). This design is sometimes called *parallel projection architecture*. To make this intuition more clear, let us first look at the syntactic component of LFG which consists of two individual projections: c(onstituent)- and f(unctional)-structure. C-structure is stated in terms of phrase structure rules and captures information about constituency and linear order. F-structure captures information about grammatical functions, such as SUBJ, OBJ (i.e., dependencies; Meurer 2017), as well as functional features such as `number` and `tense`. It is stated in terms of the quantifier-free logic of equality (Kaplan, 1989). More concretely, equality terms are *co-descriptively* added to phrase structure rules and lexical entries using the meta variables ↑ and ↓. ↑ points at the c-structure node of the mother and ↓ at the current node. Thus, The NP rule in (1), for example, states that the determiner and the noun equally contribute to the f-structure of the NP. There, the lexical entry of the determiner in (2) specifies a substructure, SPEC, subordinating the determiner to the content word.

(1)

S	→	NP (↑ SUBJ) = ↓	VP ↑=↓
NP	→	(D) ↑ = ↓	N ↑=↓
VP	→	V ↑ = ↓	AP (↑ PREDLINK) =↓

[3]Hosted at https://clarino.uib.no/iness/page (Rosén et al., 2012).

(2)

the	D	(↑ SPEC PRED) = 'the' %f_t = (GF+ ↑) $\lambda P.\lambda Q.\exists x[P(x) \wedge Q(x)]$: $(\uparrow_e \multimap \uparrow_t) \multimap (\uparrow_e \multimap$ %$f_t) \multimap$ %f_t
cat	N	(↑ PRED) = 'cat' $\lambda x.cat(x)$: $\uparrow_e \multimap \uparrow_t$
is	V	(↑ PRED) = 'be<(↑ PREDLINK)>(↑ SUBJ)' (↑ PREDLINK SUBJ) = (↑ SUBJ)
fast	A	(↑ PRED) = 'fast<(↑ SUBJ)>' $\lambda x.fast(x)$: $(\uparrow \text{SUBJ})_e \multimap \uparrow_t$

Generally, by resolving equalities, meta variables pointing at individual c-structure nodes are resolved to f-structure indices (many-to-one mapping). This process is visualized in Figure 2. As the figure indicates, the dependencies that LFG's f-structure captures are more articulated than classic dependencies as they can share structures across different PREDs, as annotated in the lexical entry for *is*. For a more comprehensive comparison involving further differences between f-structure and dependencies see Haug (2023).

2.2 Glue semantics

LFG's Glue semantics (Asudeh, 2023) specifies meaning representations, called meaning constructors (MCs).[4] They can be defined in two ways: co-descriptively, i.e., in parallel to c- and f-structure information in the lexicon (highlighted in blue in (2)) and phrase structure, or via description-by-analysis, which takes an assembled f-structure as input and rewrites it into a semantic representation. As we propose a hybrid approach (Wedekind and Kaplan, 1993), this warrants further explanation.

Description-by-analyis (DBA) rules provide independent way of introducing meaning constructors to syntactic representations, here f-structures (e.g, Andrews 2010).[5] Compared to co-descriptive semantics, they are more suitable to capture variation in the immediate syntactic and semantic context that affects semantic interpretation (Zymla, 2017). For example, the comparative complementizer *than* in Figure 2 is interpreted differently de-

[4]A more comprehensive introduction can be found in Dalrymple (1999). See also Dalrymple et al. (1993).

[5]DBA is technically a framework agnostic way of introducing meaning constructors that can be applied to different types of syntactic representations. Notably, It has been applied to Universal Dependency parses Findlay et al. (2023); Zymla (2018).

pending on its complement (e.g., elided VP vs overt VP). A simplified rule that produces the MC used in Figure 2 is illustrated in Figure 1. There, `#f...#j` are variables over f-structure nodes which are related via the given relation labels. Thus, the left-hand side can be understood as a query searching for a matching graph structure. Given that it matches the f-structure in Figure 2, the MC on the right is added to the premise set, essentially enabling the interpretation of the comparative clause.[6]

```
#f PREDLINK #h SUBJ #g & #h DEGREE 'comparative'
& #h ADJUNCT #a in_set #o OBL-COMP #i OBJ #j ==>
    #i GLUE
```

$$\lambda R.\lambda x.\lambda y.\exists\delta[R(\delta)(x) \wedge \neg R(\delta)(y)] : (\#g_d \multimap \#g_e \multimap \#f_t) \multimap (\#g_e \multimap \#j_e \multimap \#f_t).$$

Figure 1: DBA-rule for the comparative construction

MCs separate the logic of composition (linear logic; Girard 1995) and meaning language, allowing for some flexibility in the choice of the latter. Composition is resource-sensitive and flexible.[7] The Curry-Howard isomorphism (Curry et al., 1958; Howard, 1980) postulates parallels between lambda calculus operations and deduction processes in linear logic proofs. Thus, example (3) draws the parallel between function application and implication ($\multimap$) elimination, and example (4) describes the parallels between lambda abstraction and implication introduction. Consequently, Glue semantics is compatible with any meaning language whose combinatorial possibilities can be stated in λ-terms. We use λ-FOL (first-order logic) and λ-DRT (discourse representation theory; Kamp and Reyle 1993) to illustrate this.

(3) $$\frac{f : A \multimap B \qquad a : A}{f(a) : B} \multimap_E$$

(4) $$\frac{\begin{array}{c}[x : A]^i \\ \vdots \\ f(x) : B\end{array}}{\lambda x.f(x) : A \multimap B} \multimap_{I,i}$$

[6] The first argument of the comparative semantics is the semantic contribution of the adjective. However, the rule requires a predicate with a degree variable which is not provided by the lexical entry in (2). Reconciling this mismatch is discussed in section 4.

[7] Linear logic is commutative and non-associative by default. According to Moot and Retoré (2012), this is too flexible. However, this issue has been at least partially addressed in, e.g., Lev (2007); Findlay and Haug (2022); Zymla (2024) from a computational perspective. We do not explore this point further in this paper.

2.3 Reasoning

Reasoning based on XLE's LFG grammars has been pursued, for example, by Bobrow et al. (2007) and Lev (2007). These two works represent two general approaches respectively: i) reasoning via rewriting with a focus on intensional semantics (see also, e.g., Condoravdi et al. 2001), and ii) reasoning with theorem provers. The work presented here aligns with Lev's (2007) approach. Furthermore, it is inspired by more recent work in formal computational semantics by Haruta et al. (2020, 2022), who compose meanings via categorical grammar parsers and use the Vampire theorem prover to prove inferences by refutation (Kovács and Voronkov, 2013). As the focus of this paper is the educational value of computational tools based on formal linguistics, we also draw parallels to Blackburn and Bos (2005), who developed a conversational agent, CURT (clever use of reasoning tools), and highlighted how reasoning may affect dialogue interactions.

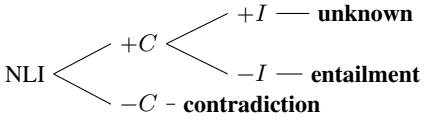
Consequently, two categories of reasoning can be considered: reasoning for natural language inference (NLI) and reasoning in dialogue. The first one is aptly exemplified by the Fracas testsuite (Cooper et al., 1996), which consists of examples like:

(5)
A Swede won a Nobel prize.
Every Swede is a Scandinavian.

Did a Scandinavian win a Nobel prize?
→ YES

As (5) shows, NLI examples consist of one or more premises, a conclusion, and a label corresponding to the entailment status of the conclusion (YES, NO, Don't know; MacCartney and Manning 2009). The goal of the task is to predict the correct label.

As part of their dialogue system, Blackburn and Bos (2005) establish informativity (+/-I) and consistency (+/-C) checks as essential tasks for reasoning in dialogue.[8] The NLI task can also be broken down to consist of these two tasks:

(6)


[8] While these checks are too strict to model the nuances of real world data, they provide useful insights into the incremental tracking of (shared) knowledge in dialogue settings.

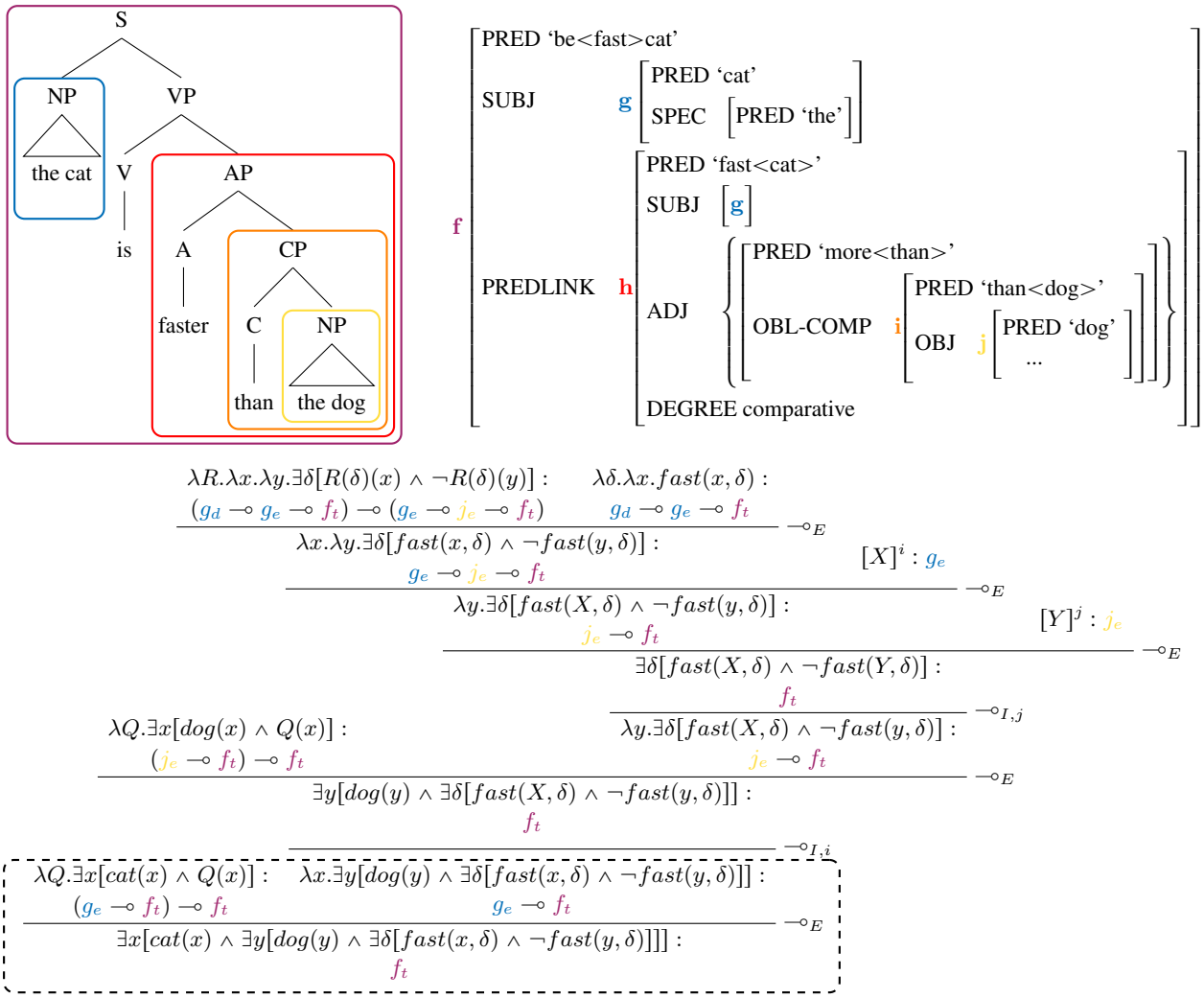


Figure 2: **LFG example derivation:** *The cat is faster than the dog.* This example illustrates the modular representation of c-structure, f-structure, and compositional semantics in LFG. C-structure captures linear order and constituency. F-structure abstracts away from surface form via a many-to-one mapping from c- to f-structure nodes. The semantics use f-structure indices as linear logic resources to encode combinatory possibilities which are stated in terms of a prooftree. Generally, hierarchical structures are broken down and re-assembled.

3 Computational implementation

This section presents our system that computationally implements the pipeline described in Figure (3). The main innovation presented in this paper is the use of LiGER (Linguistic graph expansion and rewriting) to mediate between syntax, compositional semantics and reasoning. Furthermore, we put some focus on the Blackburn and Bos (2005)-style interface to the Vampire theorem prover. However, we also briefly discuss the contribution of the other components.

3.1 Parsing via XLE+Glue

We build on computational Glue resources that have been developed in the past few years, primarily, the Glue semantics workbench (GSWB; Meßmer and Zymla 2018), a Glue prover heavily inspired by Lev's (2007) work, building on Hepple (1996), and an interface to XLE called XLE+Glue (Dalrymple et al., 2020).[9] Generally, the current main use of these tools is the verification of analyses with a focus on semantics and its interfaces, e.g., Przepiórkowski and Patejuk (2023). Recently, Butt et al. (2024) have presented a system that allows for the incorporation of prosodic information to disambiguate semantic analyses of questions in Urdu, thus, covering the full pipeline from speech signal to semantic parsing.

[9]While XLE itself is available only under a restrictive license, the semantic resources developed for it are all open source, including the new tools presented in this paper.

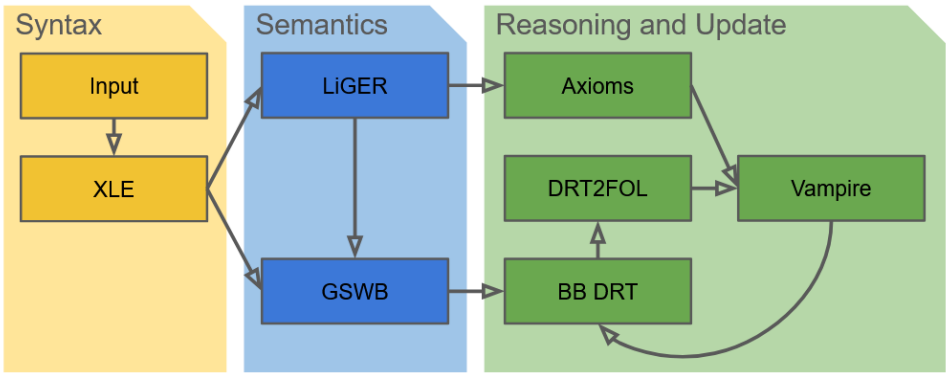


Figure 3: **The XLE+Glue pipeline:** three modular systems cover syntax, semantics, and reasoning. The syntax is specified in terms of LFG grammars written in the XLE. The XLE also specifies a core semantics that can be enriched and contextualized via LiGER. The GSWB calculates DRT-style meaning representations following Glue semantics principles. The DRSs are resolved in the reasoning module and translated into FOL for reasoning. LiGER optionally contributes additional axioms which are triggered by specific syntactic and semantic configurations to ensure correct reasoning.

.

The work presented in this paper is based on a newly developed Grammar for English that covers part of the Fracas testsuite, particularly, the section on quantifiers and the section on adjectives. Without going into detail, this grammar makes use of the various tools that XLE provides to develop larger-scale grammars, particularly, morphological analyzers, templates, parameterized rules, and more. To constrain ambiguities it makes use of OT-marks (loosely based on optimality theory). In terms of the presented syntactic analyses, it closely follows the large English ParGram grammar and Butt et al. (1999), but is extended with a co-descriptive semantics (cf. examples (1) and (2)). The semantics are resolved by the GSWB which generates Boxer-style DRT representations via a simple interface to Blackburn and Bos's Prolog code.[10]

3.2 Simple reasoning

For reasoning, DRSs are translated into first-order logic and then to the TPTP format (Sutcliffe et al., 2006). We built a Python interface that queries the Vampire theorem prover (Kovács and Voronkov, 2013) with positive and negative consistency and informativity checks. The positive checks rely on model building rather than satisfiability checking. Vampire supports this in addition to several other proof search strategies, but is mainly geared towards finding proofs via refutation.

(7) For some (set of conjoined) premise(s) p and a hypothesis q:

a.	$\neg(p \rightarrow q)$	+informative
b.	$p \rightarrow q$	-informative
c.	$p \wedge q$	+consistent
d.	$p \rightarrow \neg q$	-consistent

We extract meaningful labels from the Vampire output to present to users, concretely: the termination reason, whether a finite model was found, and the *SZS status* (Sutcliffe, 2008).[11] From these labels, we heuristically determine the success of the individual checks, and, consequently, the status with respect to the NLI task (cf. example (6)).

3.3 Extended reasoning mediated by LiGER

The system so far is essentially a re-implementation of Blackburn and Bos (2005) modeling the syntax/semantics interface in a different manner. While we believe that this has merits in its own right (particularly, the modularization of syntax and semantics), we extend its coverage with a principled approach to tackling more complex reasoning problems, such as those presented in Haruta et al. (2022). The key tool for this is LiGER which allows for the specification of rewrite rules to apply to f-structures. It plays two roles: i) non-invasively extending the semantics, and ii) determining relevant axioms needed for correct reasoning.

[10] Various aspects of the code have been adapted in accordance with the GNU license. The original files are available at https://www.let.rug.nl/bos/comsem/software2.html.

[11] Vampire's termination reason describes its result which also includes technical reasons, e.g., timeouts, whereas the SZS status focuses on the outcome of the reasoning process. Although there generally is a clear mapping from termination reason to SZS status, we use both for maximal informativity.

The first role is simply a rendering of the description-by-analysis idea presented in section 2.2. The important conceptual idea here is that the base grammar is self-sufficient, i.e., it produces semantic representations which can be optionally extended via DBA (see section 4).

The second role is inspired by Bobrow et al. (2007), who use DBA as an interface to external resources to enrich semantic representations. Concretely, we re-model Haruta et al.'s (2022) system for interpreting gradable adjectives and generalized quantifiers (a Python interface between a CCG parser and Vampire) into a system that is extensible and variable without the need to understand a complex code structure. To illustrate this, let us look at a core component of Haruta et al.'s (2022) analysis of gradable adjectives, particularly, their comparative use: the consistency postulate.

(8) (CP) $\forall x \forall y[\exists \delta[A(x,\delta) \wedge \neg A(y,\delta)] \rightarrow \forall \delta[A(y,\delta) \rightarrow A(x,\delta)]]$,
where A is an arbitrary gradable adjective.
(Haruta et al., 2022, p. 148)

The axiom in (8) is not intuitively part of the compositional semantics of an utterance, but rather is required for the semantics of gradable adjectives to fall out correctly. However, it essentially requires quantification over properties (indicated by the use of the variable A). This is accounted for in terms of a DBA rule of the following kind:

```
#a PRED %adj #a DEGREE 'comparative' ==>
  #a AXIOM   ∀x∀y[∃δ[%adj(x,δ) ∧ ¬%adj(y,δ)]
             → ∀δ[%adj(y,δ) → %adj(x,δ)]].
```

Figure 4: DBA-rule for extracting axioms

In essence, this (simplified) rule introduces an axiom based on the presence of an adjective with a comparative form and generates a CP axiom for that adjective. This information is integrated with the call to Vampire, as it affects how the prover is called. Concretely, reasoning about degrees following Haruta et al. (2022) requires arithmetic reasoning, a non-finite domain that eliminates model building as a proof search strategy.

In summary, LiGER is used on two fronts to extend the expressiveness of the compositional semantics and to trigger the axioms required to maintain correct results during inference. The LiGER output also affects the interface to Vampire directly to account for different input requirements and outputs for different kinds proofs.

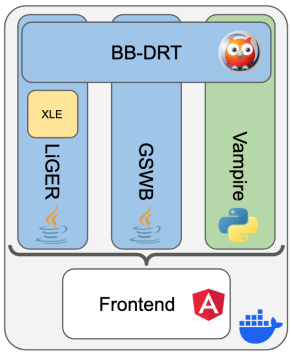


Figure 5: **System architecture:** a modularized service architecture that is accessed via a browser-based user interface.

3.4 Technical details

The whole system is couched in a modular service architecture, where individual modules are deployed in Docker containers. A browser-based application allows users to access the containers and links their functionalities (see Figure 5). The LiGER container also provides a lightweight interface to XLE.[12] This architecture enables cross-platform use of the system and minimizes the need for technical know-how. The Prolog code for BB-DRT is not deployed in a separate container but is copied across containers as it is relatively lightweight, reducing traffic across containers.

3.5 User interface and visualization

The system provides seperate interfaces to i) *parsing*, ii) *regression testing*, and iii) *inference*. Number i) and iii) are (partially) illustrated in Figure 7. The inference interface is inspired by Blackburn and Bos's (2005) conversational agent CURT. It provides access to the conversation history with the possibility to prune it. Furthermore, it allows the manual specification of axioms to test their effect on reasoning (not in the picture).

We use a glyph to optionally relay the detailed results of the inference checks to users. This is illustrated in Figure 6. The example there is an instance of a contradiction, as indicated by the refuted positive consistency check and successful negative consistency check. Although this makes informativity checks obsolete according to example (6), the glyph always displays all checks, highlighting the interplay between consistency and informativity.

[12] For licensing reasons, the XLE is not packed with the system but needs to be acquired independently and added to the repository.

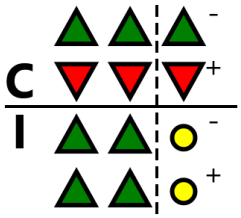


Figure 6: **Reasoning result glyph:** Green triangles indicate positive results (satisfiability), red triangles indicate negative results (refutation), and yellow symbol indicate unknown results (timeout/saturation). The glyph is horizontally separated into consistency (C) checks and informativity (I) checks with negative and positive polarity respectively. The vertical line separates satisfiability checks (left) and model building attempts (right).

Generally, the various interfaces are designed with explorative use in mind but they also enable incremental development of analyses with the regression testing interface, which is tailored towards developing new description-by-analysis rules. This supports the development of larger grammars.

4 Worked examples

We will now elaborate on the semantics we assume. This includes simpler cases including reasoning about properties and relations, but also more complex cases for which we employ a version of degree semantics following Haruta et al. (2022).

4.1 Basic semantic assumptions

Our semantics are based on a Neo-Davidsonian event semantics rendered in DRT. The first worked example is shown in Figure 7, demonstrating the correct reasoning for the problem in example (5).[13]

Due to quantifier ambiguity, the second hypothesis of (5) has two parses, presented in their equivalent FOL form in (9) and (10).[14] Here, the representation of *be* does not express anything meaningful, just that there is a *be*ing-eventuality (in the sense of Bach 1986) with two arguments.

(9) $\forall y[swede(y) \rightarrow \exists x[scandinavian(x) \wedge \exists e[be(e) \wedge arg1(e) = y \wedge arg2(e) = x]]]$

(10) $\exists x[scandinavian(x) \wedge \forall y[swede(y) \rightarrow \exists e[be(e) \wedge arg1(e) = x \wedge arg2(e) = y]]]$

[13] All analyses are laid out in detail in the appendix.

[14] Although the reading in (10) is not intuitively sensible, it does allow for the same inference. Nonetheless, this indicates the need for more fine-grained management of quantifier ambiguities, wich we leave for future work (first steps are taken in Zymla 2024).

For the inference in (5) to come out correctly, we need to add a meaning postulate (Zimmermann, 1999), as in (11).[15]

(11) $\forall x, y, e[be(e) \wedge arg1(e) = x \wedge arg2(e) = y \rightarrow x = y]$

This is to show that there is a wide range of axioms that one can consider adding to an analysis. Introducing meaning postulates via DBA-rules allows for the exploration of their impact on reasoning before integrating them into the grammar proper.

4.2 Layered analysis

In this section, we finalize the analysis of gradable adjectives following Haruta et al. (2022). However, note first that the Fracas testsuite contains several examples containing gradable adjectives that can be analyzed as simple properties, such as (12). Here, the challenge rather lies in modeling the syntax/semantics interface correctly to capture the modifying nature of the relative clause (e.g., Heim and Kratzer 1998).

(12) Some great tenors are Swedish.
Are there great tenors who are Swedish?

This extends to examples with more complicated constructions like the superlative:

(13) An Italian became the world's greatest tenor.
Was there an Italian who became the world's greatest tenor?

Thus, in many cases, reasoning via pattern matching is sufficient: as the noun phrases perfectly match, their exact semantics become less relevant.

However, often, gradable adjectives are challenging for automated reasoning because they are highly context sensitive. To make this intuition clear, first consider example (14), which illustrates the context sensitivity of the adjective *large*, whose meaning is mediated by the immediate context, here the modified noun. Generally, positive instances of gradable adjectives are evaluated based on contextually determined *comparison classes*.[16]

[15] We use a classic analysis of *be* also used in Blackburn and Bos (2005), which could also be stated directly in the semantics.

[16] The examples in the Fracas testsuite determine comparison classes in the immediate linguistic context, but comparison classes may be determined by the wider context, including extralinguistic cues (e.g., Kennedy and McNally 2005).

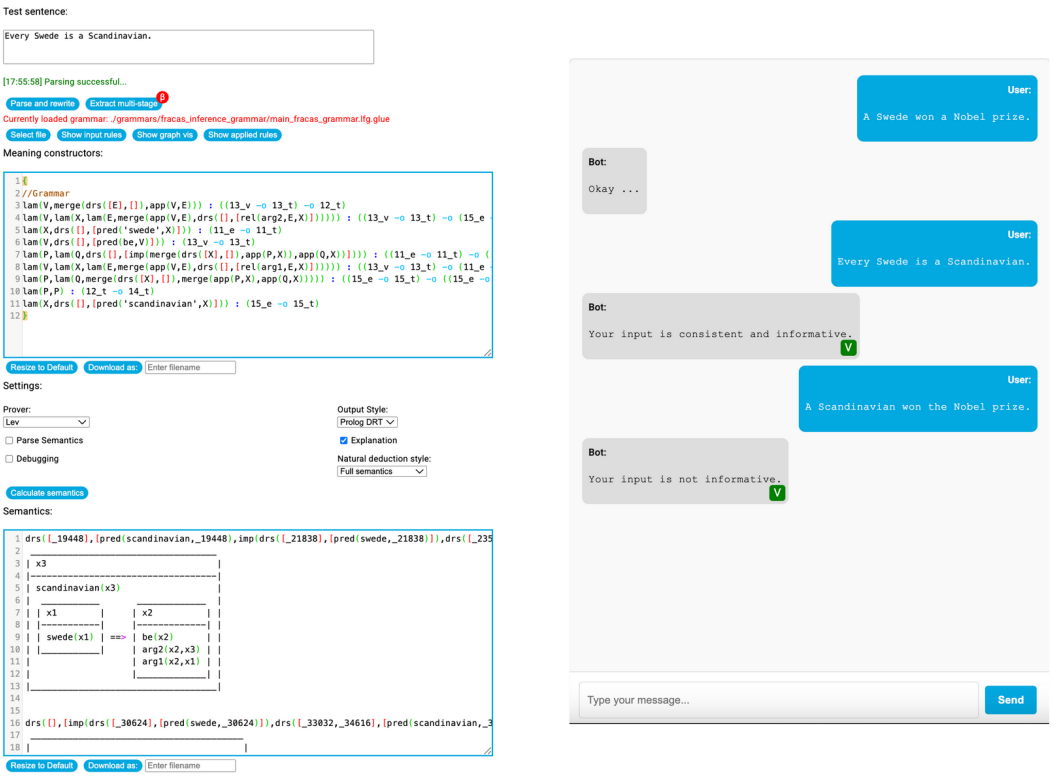


Figure 7: **XLE+Glue browser-based user interface:** On the left, the parsing interface is highly customizable and allows for the exploration of the compositional semantics underlying a parse. On the right, the chat interface allows for testing of the inference capabilities.

(14)
All mice are small animals.
Mickey is a large mouse.

Is Mickey a large animal? → NO

By making adjectives sensitive to a comparison class, the two different meanings of *large* in (14) can be explained. Concretely, we want to express that Mickey surpasses the threshold for a large mouse but not the threshold for a large animal. However, first we need to type-raise *large* to become a degree predicate.[17] The two steps can be encoded in terms of DBA-rules that extend the compositional semantics, as shown in Figure 8.

```
#a PRED %adj & #a DEGREE %d ==>
#a GLUE   λP.λδ.λx.%adj(x, δ) :
          (#a_e ⊸ #a_t) ⊸ (#a_d ⊸ #a_e ⊸ #a_t).

#n PRED %pred & #n ADJ #a PRED %adj &
#a DEGREE 'positive' ==>
#a GLUE   λP.λx.∃δ[δ > θ_%adj(%pred) ∧ P(δ)(x)]
          (#a_d ⊸ #a_e ⊸ #a_t) ⊸ #a_e ⊸ #a_t .
```

Figure 8: DBA-rule for positive gradable adjectives

In addition to the CP (see example (8)), we need to further specify the meaning of *large* a *positive* adjective, and *small*, a *negative* adjective:

(15)
(up) $\forall x, \delta'[large(x, \delta') \rightarrow \forall \delta''[\delta'' \leqslant \delta' \rightarrow large(x, \delta'')]]$
(down) $\forall x, \delta'[small(x, \delta') \rightarrow \forall \delta''[\delta' \leqslant \delta'' \rightarrow small(x, \delta'')]]$

These axioms say that if something is *large* to some degree δ it is also large to any degree smaller than that. Conversely, if something is small to a degree δ, it is also small to any larger degree.[18] Together with axioms for the antonym relation between *large* and *small*, the inference in (14) succeeds. An appropriate DBA-rule generalizes over positive and negative adjectives accordingly.

The type-raising rule also resolves the mismatch mentioned in footnote 6, allowing us to deal with comparatives as in (16) (see Appendix A.5).

[17] Accordingly, we have to slightly change the rule in Figure 1, which we simplified for sake of exposition. We accept this extra step to preserve the integrity of the core grammar.

[18] A reviewer points out that Haruta et al. (2022) show that the CP, (8), follows from *up* and *down* questioning the necessity of these axioms. However, the inverse is not true. Thus, they are required for cases like (14) which do not contain explicit comparatives (see Appendix A.4).

(16)
The PC-6082 is faster than every ITEL computer.
The ITEL-ZX is an ITEL computer.

Is the PC-6082 faster than the ITEL-ZX? → YES

5 Limitations

While the present system is developed in a task-oriented fashion, it suffers from the usual drawbacks of formal computational linguistics, such as a lack of robustness (particularly, with respect to unseen data), and tedious ambiguity management (Bunt, 2008), particularly as one attempts to scale up grammars (Flickinger et al., 2017). Thus, the present system should not be seen as ready for real NLP applications (yet). Nonetheless, it contributes to closing the gap between formal and computational linguistics, by making the latter more accessible to practitioners of the former, which should be mutually beneficial for both disciplines (e.g., Bender 2008; King 2016). Furthermore, through regression testing (Chatzichrisafis et al., 2007), the grammar presented here, as well as the system as a whole, are continuously expanded.

Although we see the modular architecture of LFG as a benefit regarding the explainability of different aspects of language that affect semantic interpretation, the reliance on XLE can be a drawback. To address this, we also provide an integration of the semantics tool with Stanza's dependency parser (Qi et al., 2020).[19] However, we do not yet provide a reasonably-sized set of semantic rules ready for inference testing. This is an avenue for future work.

6 Summary

This paper presents an open source computational resource that enables the exploration of computational semantics and reasoning through the lense of LFG's Glue semantics. Its hallmarks are i) an interface to the Vampire theorem prover, ii) a principled system for exploring formal semantics and their use in automated reasoning at various levels of complexity, and iii) a grounding in the seminal work on formal approaches to natural language inference by Blackburn and Bos (2005). These hallmarks come with various avenues for future work. On the LFG-side, Butt et al. (2024) integrate prosodic information into a fully formal system for semantic parsing, thus, enabling a comprehensive exploration of formal linguistic insights from the speech signal to reasoning. Linking their work with present resources would grant an even deeper understanding of the interplay between syntax, prosody, and semantic interpretation.

On the reasoning side, the present work not only allows users to explore Blackburn and Bos (2005), but also extends to their (unpublished) material on discourse representation theory, enabling, for example, the exploration of anaphora and presuppositions. Furthermore, this paves the way for the exploration of discourse relations (Asher and Lascarides, 2003) further down the line.

More speculatively, we believe that aiming for a hybrid system, where machine-learning methods are used to intervene at various levels of linguistic analysis (syntax, semantics, pragmatics) could be mutually beneficial, potentially increasing the explainability of large language models (as a representative of the most prevalent machine learning methods in current NLP), but, importantly also improving the robustness of the present system, e.g., by helping with disambiguation and possibly by modulating the reasoning process.

All in all, we present a principled and thus instructive way to explore formal semantics and reasoning. The system can be locally deployed as a browser app with an accessible user interface, making it interesting for a broad audience within computational linguistics and adjacent fields.

[19] https://github.com/Mmaz1988/xleplusglue/tree/2025_xleplusud

Acknowledgments

We thank the German Research Foundation (DFG) for financial support within the project D02 of the SFB/Transregio 161.

References

Avery D. Andrews. 2010. Propositional Glue and the correspondence architecture of LFG. *Linguistics and Philosophy*, 33(3):141–170.

Nicholas Asher and Alex Lascarides. 2003. *Logics of Conversation*. Cambridge University Press.

Ash Asudeh. 2022. Glue semantics. *Annual Review of Linguistics*, 8:321–341.

Ash Asudeh. 2023. Glue semantics. In *Handbook of Lexical Functional Grammar*, pages 651–697. Language Science Press, Berlin.

Emmon Bach. 1986. The algebra of events. *Linguistics and Philosophy*, 9(1):5–16.

Emily M. 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, Stanford, CA. CSLI Publications.

Patrick Blackburn and Johannes Bos. 2005. *Representation and inference for natural language: A first course in computational semantics*. Center for the Study of Language and Information, Stanford, CA.

Daniel G. Bobrow, Bob Cheslow, Cleo Condoravdi, Lauri Karttunen, Tracy Holloway King, Rowan Nairn, Valeria de Paiva, Charlotte Price, and Annie Zaenen. 2007. PARC's bridge and question answering system. In *Proceedings of the GEAF 2007 Workshop*, pages 1–22.

Kersti Börjars. 2020. Lexical-functional grammar: An Overview. *Annual Review of Linguistics*, 6:155–172.

Harry Bunt. 2008. Semantic underspecification: Which technique for what purpose? In *Computing Meaning*, pages 55–85. Springer.

Miriam Butt, Tina Bögel, Mark-Matthias Zymla, and Benazir Mumtaz. 2024. Alternative questions in urdu: From the speech signal to semantics. In *Proceedings of the LFG'24 Conference*, Konstanz. PubliKon.

Miriam Butt, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. The parallel grammar project. In *Proceedings of the 2002 Workshop on Grammar Engineering and Evaluation*, volume 15, pages 1–7. Association for Computational Linguistics.

Miriam Butt, Tracy Holloway King, María-Eugenia Niño, and Frédérique Segond. 1999. *A Grammar writer's cookbook*. CSLI Publications.

Nikos Chatzichrisafis, Dick Crouch, Tracy Holloway King, Rowan Nairn, Manny Rayner, and Marianne Santaholma. 2007. Regression testing for grammar-Based systems. In *Proceedings of the Grammar Engineering Across Frameworks (GEAF07) Workshop*, pages 128–143, Stanford, CA. CSLI Publications.

Cleo Condoravdi, Dick Crouch, John Everett, Valeria Paiva, Reinhard Stolle, Danny Bobrow, and Martin van den Berg. 2001. Preventing existence. In *Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001*, FOIS '01, pages 162–173, New York, NY, USA. ACM.

Robin Cooper, Dick Crouch, Jan van Eijck, Chris Fox, Josef van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, and Steve Pulman. 1996. Using the framework. Technical Report LRE 62-051 D-16, The FraCaS Consortium.

Dick Crouch, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, John T. Maxwell III, and Paula Newman. 2017. *XLE documentation*. Palo Alto Research Center.

Haskell Brooks Curry, Robert Feys, William Craig, J. Roger Hindley, and Jonathan P. Seldin. 1958. *Combinatory logic*, volume 1. North-Holland Amsterdam.

Mary Dalrymple. 1999. *Semantics and syntax in lexical functional grammar: The resource logic approach*. The MIT Press, Cambridge, MA.

Mary Dalrymple, editor. 2023. *Handbook of Lexical Functional Grammar*. Number 13 in Empirically Oriented Theoretical Morphology and Syntax. Language Science Press, Berlin.

Mary Dalrymple, John Lamping, and Vijay Saraswat. 1993. LFG semantics via constraints. In *Proceedings of the Sixth Conference on European Chapter of the Association for Computational Linguistics (EACL '93)*, page 97–105, USA. Association for Computational Linguistics.

Mary Dalrymple, Agnieszka Patejuk, and Mark-Matthias Zymla. 2020. XLE+Glue – a new tool for integrating semantic analysis in XLE. In *Proceedings of the LFG'20 Conference*, pages 89–108, Stanford, CA. CSLI Publications.

Jamie Y. Findlay and Dag T. T. Haug. 2022. Managing scope ambiguities in Glue via multistage proving. In *Proceedings of the LFG'22 Conference*, pages 144–163, Konstanz, Germany. PubliKon.

Jamie Y Findlay, Saeedeh Salimifar, Ahmet Yıldırım, and Dag T. T. Haug. 2023. Rule-based semantic interpretation for Universal Dependencies. In *Proceedings of the Sixth Workshop on Universal Dependencies (UDW, GURT/SyntaxFest 2023)*, pages 47–57.

Dan Flickinger, Stephan Oepen, and Emily M. Bender. 2017. Sustainable development and refinement of complex linguistic annotations at scale. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, pages 353–377. Springer, Dordrecht.

Jean-Yves Girard. 1995. Linear logic: Its syntax and semantics. *London Mathematical Society Lecture Note Series*, pages 1–42.

Izumi Haruta, Koji Mineshima, and Daisuke Bekki. 2020. Logical inferences with comparatives and generalized quantifiers. In *Proceedings of the 58th Annual Meeting of the ACL: Student Research Workshop*, pages 263–270, Online. ACL.

Izumi Haruta, Koji Mineshima, and Daisuke Bekki. 2022. Implementing natural language inference for comparatives. *Journal of Language Modelling*, 10(1).

Dag T. T. Haug. 2023. LFG and dependency grammar. In Mary Dalrymple, editor, *Handbook of Lexical Functional Grammar*, chapter 43, pages 1829–1859. Language Science Press, Berlin.

Irene Heim and Angelika Kratzer. 1998. *Semantics in generative grammar*. Blackwell, Oxford, UK.

Mark Hepple. 1996. A compilation-chart method for linear categorial deduction. In *Proceedings of the 16th Conference on Computational Linguistics*, volume 1, pages 537–542. Association for Computational Linguistics.

William A. Howard. 1980. The formulae-as-types notion of construction. *To HB Curry: Essays on Combinatory Logic, Lambda calculus, and Formalism*, 44:479–490.

Hans Kamp and Uwe Reyle. 1993. *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and Discourse Representation Theory*, volume 42 of *Studies in Linguistics and Philosophy*. Kluwer Academic Publishers, Dordrecht/Boston.

Ronald M. Kaplan. 1989. The formal architecture of Lexical-Functional Grammar. *Journal of Information Science and Engineering*, 5(4):305–322.

Ronald M. Kaplan and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Mary Dalrymple, Ronald M. Kaplan, John T. Maxwell III, and Annie Zaenen, editors, *Formal Issues in Lexical-Functional Grammar*, pages 1–102. Stanford University, Stanford, CA.

Christopher Kennedy and Louise McNally. 2005. Scale structure, degree modification, and the semantics of gradable predicates. *Language*, pages 345–381.

Tracy Holloway King. 2016. Theoretical linguistics and grammar engineering as mutually constraining disciplines. In *Proceedings of the Joint 2016 Conference on Head-driven Phrase Structure Grammar and Lexical Functional Grammar, Polish Academy of Sciences, Warsaw, Poland*, pages 339–359, Stanford, CA. CSLI Publications.

Laura Kovács and Andrei Voronkov. 2013. First-order theorem proving and vampire. In *International Conference on Computer Aided Verification*, pages 1–35. Springer.

Iddo Lev. 2007. *Packed computation of exact meaning representations*. Ph.D. thesis, Stanford University.

Bill MacCartney and Christopher D. Manning. 2009. An extended model of natural logic. In *Proceedings of the eight International Conference on Computational Semantics*, pages 140–156.

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.

Moritz Meßmer and Mark-Matthias Zymla. 2018. The Glue semantics workbench: A modular toolkit for exploring linear logic and Glue semantics. In *Proceedings of the LFG'18 Conference*, pages 249–263, Stanford, CA. CSLI Publications.

Paul Meurer. 2017. From LFG structures to dependency relations. *Bergen Language and Linguistics Studies*, 8(1).

Richard Moot and Christian Retoré. 2012. *The logic of categorial grammars: A deductive account of natural language syntax and semantics*. Number 6850 in Lecture Notes in Computer Science. Springer, Heidelberg.

Adam Przepiórkowski and Agnieszka Patejuk. 2023. Filling gaps with Glue. In *Proceedings of the LFG'23 Conference*, pages 223–240, Konstanz, Germany. PubliKon.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Victoria Rosén, Koenraad De Smedt, Paul Meurer, and Helge Dyvik. 2012. An open infrastructure for advanced treebanking. In *META-RESEARCH Workshop on Advanced Treebanking at LREC2012*, pages 22–29.

Sebastian Sulger, Miriam Butt, Tracy Holloway King, Paul Meurer, Tibor Laczkó, György Rákosi, Cheikh M Bamba Dione, Helge Dyvik, Victoria Rosén, Koenraad De Smedt, Agnieszka Patejuk, Özlem Çetinoğlu, I Wayan Arka, and Meladel Mistica. 2013. ParGramBank: The ParGram parallel treebank. In *ACL*, pages 550–560.

Geoff Sutcliffe. 2008. The SZS ontologies for automated reasoning software. In *Proceedings of the LPAR Workshops: Knowledge Exchange: Automated Provers and Proof Assistants (KEAPA 2008), and the 7th International Workshop on the Implementation of Logics (IWIL-2008)*, volume 418 of *CEUR Workshop Proceedings*, pages 38–49. CEUR-WS.org.

Geoff Sutcliffe, Stephan Schulz, Koen Claessen, and Allen Van Gelder. 2006. Using the TPTP language for writing derivations and finite interpretations. In *Automated Reasoning – IJCAR 2006*, volume 4130 of *Lecture Notes in Computer Science*, pages 67–81, Seattle, WA, USA. Springer.

Jürgen Wedekind and Ronald M. Kaplan. 1993. Type-driven semantic interpretation of f-structures. In *Proceedings of the Sixth EACL*, pages 404–411.

Thomas Ede Zimmermann. 1999. Meaning postulates and the model-theoretic approach to natural language semantics. *Linguistics and Philosophy*, 22(5):529–561.

Mark-Matthias Zymla. 2017. Comprehensive annotation of cross-linguistic variation in tense and aspect categories. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS) — Long Papers*, Montpellier, France.

Mark-Matthias Zymla. 2018. Annotation of the syntax/semantics interface as a bridge between deep linguistic parsing and TimeML. In *Proceedings 14th Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, pages 53–59.

Mark-Matthias Zymla. 2024. Ambiguity management in computational Glue semantics. In *Proceedings of the LFG'24 Conference*, pages 285–310, Konstanz, Germany. PubliKon.

A Worked examples

A.1 Example (12)

(17) a. Some great tenors are Swedish.
$\exists x, e[tenor(x) \wedge great(x) \wedge swedish(e) \wedge be(e) \wedge arg1(e) = x]$

b. There are some great tenors who are Swedish.
$\exists x, y, e_1, e_2[tenor(x) \wedge great(x) \wedge swedish(e) \wedge be(e_1) \wedge arg1(e) = x \wedge be(e_2) \wedge arg1(e_2) = y \wedge arg2(e_2) = x]$

Generated semantics:

```
 _____________
| x2 x1       |
|-------------|
| great(x2)   |
| tenor(x2)   |
| swedish(x1) |
| be(x1)      |
| arg1(x1,x2) |
|_____________|
```

```
 _____________
| x2 x4 x3 x1 |
|-------------|
| great(x2)   |
| tenor(x2)   |
| swedish(x4) |
| be(x4)      |
| arg1(x4,x2) |
| be(x1)      |
| arg1(x1,x3) |
| arg2(x1,x2) |
|_____________|
```

Used axioms:

```
fof(be_axiom, axiom,
   ![X,Y,Z] : ((be(X) & arg1(X,Y) & arg2(X,Z)) => Y = Z)).
```

Inference output:

	+consistent	-consistent	+informative	-informative
Termination reason	+	+	-	+
SZS status	+	+	-	+
Model found	+	+	-	+

A.2 Example (5)

(18) a. A Swede won a Nobel prize
$\exists x, e[swede(x) \wedge prize(y) \wedge win(e) \wedge arg1(e) = x \wedge arg2(e) = y]$

b. Every Swede is a Scandinavian
$\forall x, e[swede(x) \rightarrow \exists e[scandinavian(e) \wedge be(e) \wedge arg1(e) = x]]$

c. A Scandinavian won the Nobel prize.
$\exists x, e[scandinavian(x) \wedge prize(y) \wedge win(e) \wedge arg1(e) = x \wedge arg2(e) = y]$

Generated semantics:

```
 _____________
| x3 x2 x1    |
|-------------|
| swede(x3)   |
| prize(x2)   |
| win(x1)     |
| arg1(x1,x3) |
| arg2(x1,x2) |
|_____________|
```

```
 ___________________________________________
|                                           |
|-------------------------------------------|
|  ___________       __________________     |
| | x1        |     | x3 x2            | |
| |-----------|     |------------------| |
| | swede(x1) | ==> | scandinavian(x3) | |
| |___________|     | be(x2)           | |
|                   | arg1(x2,x1)      | |
|                   | arg2(x2,x3)      | |
|                   |__________________| |
|___________________________________________|
```

```
 _____________
| x3 x2 x1    |
|-------------|
| scandi(x3)  |
| prize(x2)   |
| win(x1)     |
| arg1(x1,x3) |
| arg2(x1,x2) |
|_____________|
```

Used axioms:

```
fof(be_axiom, axiom,
    ![X,Y,Z] : ((be(X) & arg1(X,Y) & arg2(X,Z)) => Y = Z)).
```

Inference output:

	+consistent	-consistent	+informative	-informative
Termination reason	+	+	-	+
SZS status	+	+	-	+
Model found	+	+	-	+

A.3 Example (13)

(19) a. An Italian became the greatest tenor.
$\exists x, y, e[italian(x) \wedge great(y) \wedge tenor(y) \wedge become(e) \wedge arg2(e) = x \wedge arg1(e) = y]$

b. There was an Italian who became the greatest tenor.
$\exists x, y, e[italian(x) \wedge great(y) \wedge tenor(y) \wedge become(e) \wedge arg1(e) = x \wedge arg2(e) = y]$

```
 _____________
| x2 x3 x1    |
|-------------|
| italian(x2) |
| great(x3)   |
| tenor(x3)   |
| become(x1)  |
| arg2(x1,x3) |
| arg1(x1,x2) |
|_____________|
```

```
 ________________
| x3 x5 x4 x2 x1 |
|----------------|
| italian(x3)    |
| great(x5)      |
| tenor(x5)      |
| become(x4)     |
| arg2(x4,x5)    |
| arg1(x4,x3)    |
| be(x1)         |
| arg2(x1,x3)    |
| arg1(x1,x2)    |
|________________|
```

Used axioms:

```
fof(be_axiom, axiom,
    ![X,Y,Z] : ((be(X) & arg1(X,Y) & arg2(X,Z)) => Y = Z)).
```

Inference output:

	+consistent	-consistent	+informative	-informative
Termination reason	+	+	-	+
SZS status	+	+	-	+
Model found	+	+	-	+

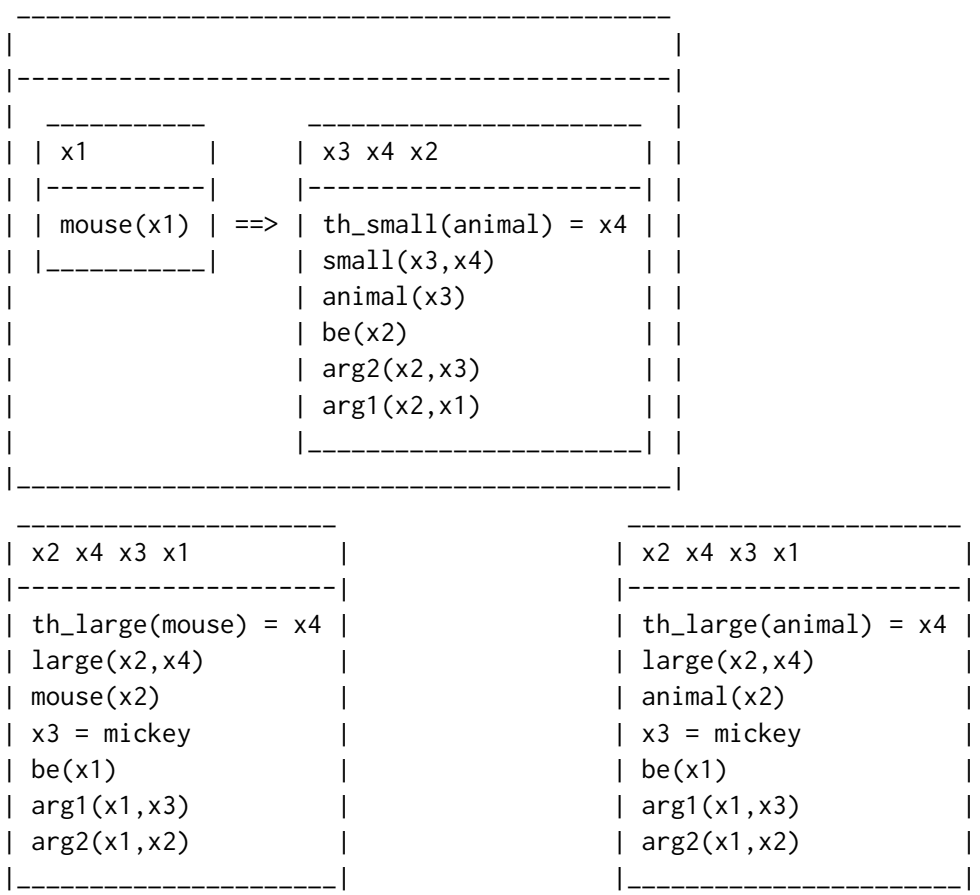
A.4 Example (14)

(20) a. All mice are small animals.
$\forall x[mouse(x) \rightarrow \exists y, \delta, e[animal(y) \wedge small(y, \delta) \wedge \theta_{\texttt{small}}(animal) = \delta \wedge be(e) \wedge arg1(e) = x \wedge arg2(e) = y]$

b. Mickey is a large mouse.
$\exists x, y, \delta, e[x = Mickey \wedge mouse(y) \wedge large(y, \delta) \wedge \theta_{\texttt{large}}(mouse) = \delta \wedge be(e) \wedge arg1(e) = x \wedge arg2(e) = y]$

c. Mickey is a large animal.
$\exists x, y, \delta, e[x = Mickey \wedge animal(y) \wedge large(y, \delta) \wedge \theta_{\texttt{large}}(animal) = \delta \wedge be(e) \wedge arg1(e) = x \wedge arg2(e) = y]$

Generated semantics:



Used axioms:

```
% adjectives
tff(large_type, type, large: ($i * $int) > $o).
tff(small_type, type, small: ($i * $int) > $o).

%comparison classes
tff(large_cc_type, type, th_large: $i > $int).
tff(small_cc_type, type, th_small: $i > $int).

%predicatives
tff(be_type, type, be: $i > $o).
tff(arg1_type, type, arg1: ($i * $i) > $o).
tff(arg2_type, type, arg2: ($i * $i) > $o).

%nouns
tff(mouse_type, type, mouse: $i > $o).
tff(animal_type, type, animal: $i > $o).

%names
tff(pn_type1, type, mickey: $i).
tff(pn_type2, type, minni: $i).
tff(pn_type3, type, animal: $i).
tff(pn_type4, type, mouse: $i).

%predicative meaning postulate
tff(axiom1,axiom,(![A : $i]: (![B : $i]: (![C : $i]: ((be(A) & (arg1(A,B) & arg2(A,C))) => (B = C))))
    )).

%from events to adjectives
tff(axiom2,axiom,(![A : $i]: (![B : $i]: (![C : $int]: ((arg1(A,B) & large(A,C)) => large(B,C)))))).
tff(axiom3,axiom,(![A : $i]: (![B : $i]: (![C : $int]: ((arg1(A,B) & small(A,C)) => small(B,C)))))).

tff(axiom4,axiom,(![A : $i]: (?[B : $int]: (large(A,B) & ~ (?[C : $int]: ($greater(C,B) & large(A,C))
```

```
    )))))).
tff(axiom5,axiom,(![A : $i]: (![B : $int]: (large(A,B) <=> (![C : $int]: ($lesseq(C,B) => large(A,C))
    )))))).

tff(axiom6,axiom,(![A : $i]: (?[B : $int]: (small(A,B) & ~ (?[C : $int]: ($greater(B,C) & small(A,C))
    )))))).
tff(axiom7,axiom,(![A : $i]: (![B : $int]: (small(A,B) <=> (![C : $int]: ($lesseq(B,C) => small(A,C))
    )))))).

%comparison class
tff(cclass,axiom, (![D: $int, D1: $int]:((th_large(animal) = D & th_small(animal) = D1) => $less(D1,D
    )))).

%antonym
tff(antonym1, axiom, ![X: $i, D: $int]: (large(X,D) <=> ~small(X,D))).
tff(antonym2, axiom, ?[X: $i, D: $int]: (large(X,D)) <=> ?[X1: $i, D1: $int]: (small(X1,D1))).
```

Inference output:

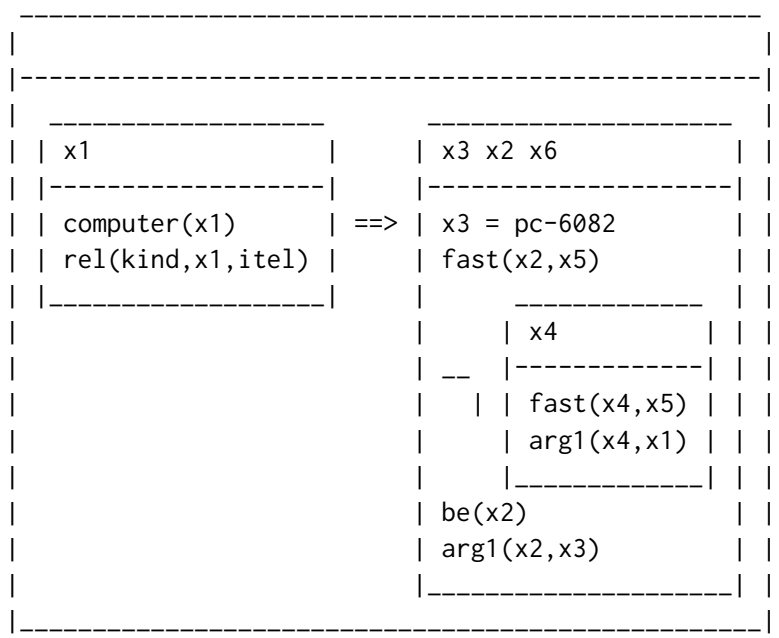
	+consistent	-consistent	+informative	-informative
Termination reason	-	?	?	?
SZS status	-	?	?	?

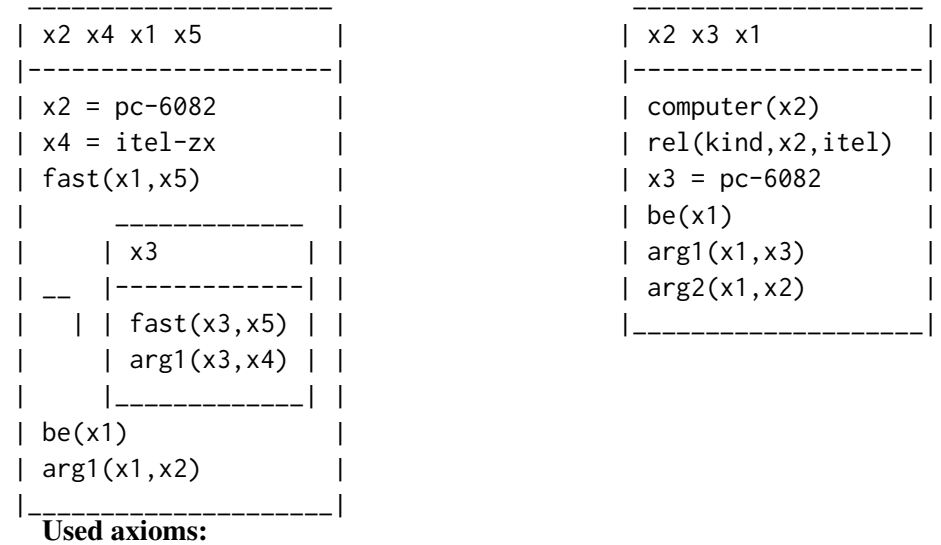
A.5 Example (16)

(21) a. The PC-6082 is faster than every ITEL computer.
$\forall y[computer(y) \wedge kind(y, itel) \rightarrow \exists x, \delta, e[fast(e, \delta) \wedge arg1(e) = x \wedge x =$ PC-6082 $\wedge$ $arg2(e) = y]]$

b. The ITEL-ZX is an ITEL computer.
$\exists x, y, e[x =$ ITEL-ZX $\wedge computer(y) \wedge kind(y, itel) \wedge be(e) \wedge arg1(e) = x \wedge arg2(e) = y]$

c. The PC-6082 is faster than the ITEL-ZX.
$\exists x, y, \delta, e[fast(e, \delta) \wedge x =$ PC-6082 $\wedge\ y =$ ITEL-ZX $\wedge\ arg1(e) = x \wedge arg2(e) = y \wedge \neg\exists e'[fast(e', \delta) \wedge arg1(e') = y]]$





Used axioms:

```
%adjectives
tff(fast_type, type, fast: ($i * $int) > $o).

%modifiers
tff(kind_type, type, kind: ($i * $i) > $o).

%predicatives
tff(be_type, type, be: $i > $o).
tff(arg1_type, type, arg1: ($i * $i) > $o).
tff(arg2_type, type, arg2: ($i * $i) > $o).

%nouns
tff(computer_type, type, computer: $i > $o).

%names
tff(pn_type1, type, 'pc-6082': $i).
tff(pn_type2, type, 'itel-zx': $i).

%predicative meaning postulate
tff(axiom1,axiom,(![A : $i]: (![B : $i]: (![C : $i]: ((be(A) & (arg1(A,B) & arg2(A,C))) => (B = C))))
    )).

%from events to adjectives
tff(axiom2, axiom, (![A: $i]: (![B: $i]: (![C: $int]: ((arg1(A,B) & fast(A,C)) => fast(B,C)))))).

tff(axiom3, axiom, (![A: $i]: (![B: $int]: (fast(A,B) <=> (![C: $int]: ($lesseq(C,B) => fast(A,C)))))
    )).
tff(axiom4, axiom, (![A: $i]: (![B: $i]: ((?[C: $int]: (fast(A,C) & ~fast(B,C))) => (![D: $int]: (
    fast(B,D) => fast(A,D)))))).
tff(axiom5, axiom, (![A: $i]: (?[B: $int]: (fast(A,B) & ~(?[C: $int]: ($greater(C,B) & fast(A,C)))))
    ).
```

Inference output:

	+consistent	-consistent	+informative	-informative
Termination reason	?	?	-	?
SZS status	?	?	-	?

- For comparatives, only a partial answer based on the refutation of the positive informativity check is given.
- ? refers to proof searches that have timed out.
- The search strategy differs in this examples as model building is not an option.

Modelling Expectation-based and Memory-based Predictors of Human Reading Times with Syntax-guided Attention

Lukas Mielczarek[*] **Timothée Bernard**[**] **Laura Kallmeyer**[*]
Katharina Spalek[*] **Benoît Crabbé**[**]

[*]firstname.lastname@uni-duesseldorf.de, Heinrich-Heine-Universität Düsseldorf
[**]firstname.lastname@u-paris.fr, Université Paris Cité

Abstract

The correlation between reading times and surprisal is well known in psycholinguistics and is easy to observe. There is also a correlation between reading times and structural integration, which is, however, harder to detect (Gibson, 2000). This correlation has been studied using parsing models whose outputs are linked to reading times. In this paper, we study the relevance of memory-based effects in reading times and how to predict them using neural language models. We find that integration costs significantly improve surprisal-based reading time prediction. Inspired by Timkey and Linzen (2023), we design a small-scale autoregressive transformer language model in which attention heads are supervised by dependency relations. We compare this model to a standard variant by checking how well each model's outputs correlate with human reading times and find that predicted attention scores can be effectively used as proxies for syntactic integration costs to predict self-paced reading times.

1 Introduction

Recently, there has been increased interest in evaluating language models (LMs) regarding their psycholinguistic plausibility, particularly in relation to two important approaches to human sentence processing: expectation-based (Hale, 2001; Levy, 2008) and memory-based theories (Gibson, 2000).

Expectation-based theories postulate that surprisal is a good indicator of human reading times (RTs), and that surprisal can be modelled with a language model. A strong correlation between surprisal and RTs was confirmed using state-of-the-art transformer-based LMs (Wilcox et al., 2023). In contrast, memory-based theories such as cue-based retrieval (Van Dyke and Lewis, 2003) explain difficulties in processing with the limitations of information encoding and retrieval in human working memory. In particular, *Dependency Locality Theory* (DLT) proposes that when processing a token, longer syntactic dependencies cause higher *integration costs* (i.e. the online cognitive cost required to integrate the token into the structure built so far), thus longer reading times (Gibson, 2000).

Against this backdrop, efforts are made to unify these approaches by constructing LMs that jointly operationalise both paradigms and generate theory-driven predictions aligned with human data. For example, Ryu and Lewis (2021) show that self-attention can be seen as cue-based retrieval. Inspired by that, Timkey and Linzen (2023) propose a unified cognitive model by training an LM with only one attention head. They observe that their model tends to attend to syntactically close tokens, resembling expected memory effects, but they do not leverage the attention patterns of their model for reading time predictions.

Linguists have produced a vast collection of work pertaining to the structures underlying language. If these theories are indeed indicative of the human cognitive process, incremental parsers such as the attach-juxtapose parser (Yang and Deng, 2020; Ezquerro et al., 2024) and the PLTAG parser (Demberg et al., 2013) should allow us to extract measures that we could link to human RTs. However, these models do not predict next token probabilities. Given the significance of the correlation between surprisal and RTs, we are interested in models that combine incremental parsing and next token prediction.

This paper approaches the question of how surprisal and structural integration costs contribute to RT predictions in two ways. We first train an LM only towards next word prediction. This LM provides surprisal, i.e. expectation-based RT predictors. We then (i) compare RT prediction based on surprisal only with RT predictions based on both surprisal and structural integration cost. We do so by obtaining surprisal from the LM and the structural costs from parsed dependency data. We observe that structural integration costs im-

Proceedings of the Second Workshop on the Bridges and Gaps between Formal and Computational Linguistics (BriGap-2), pages 52–71

prove RT prediction, which leads us to (ii) devise a dependency enhanced LM that outputs both expectation-based and memory-based processing features, which we compare in the same fashion. Again, we observe that RT predictions are improved. Finally, comparing the contributions of surprisal and structural integration costs provided in (i) and (ii), we note that the syntax-enhanced LM has a better fit to self-paced reading times while surprisal from a vanilla LM combined with parsed data is better for eye-tracking data.

In Sections 2–3, we outline our research questions and discuss related work. We follow with an investigation of natural data to establish the significance of memory-based reading time predictors (see (i) above). Finally, in Section 5 we present our dependency-enhanced neural network and investigate how well the combination of expectation-based and memory-based features from our model predicts reading times (see (ii) above).

2 Methodology

Research questions We aim to answer the following questions: [Q1] Does syntactic integration cost reflect properties of human sentence processing that are not explained by surprisal? [Q2] Can a syntax-informed language model better capture features of human sentence processing than a vanilla model, both with respect to expectation-based and memory-based costs?

We hypothesise [H1] that using syntactic integration cost improves RT predictions over a model that only includes surprisal and [H2] that small-scale transformers trained to attend to syntactic governors or dependents better reflect human language processing than their unconstrained counterparts.

Proposal To answer [Q1], we estimate the joint predictive power of surprisal and a memory-based integration cost on eye-tracking and on self-paced reading time data. The structural integration cost is in this case obtained from parse trees based on an off-the-shelf parser (silver parses). We confirm that both expectation-based and memory-based theories give rise to significant predictors for RTs and that including both aspects in a linear mixed effects model significantly improves RT predictions over including only the expectation-based predictor.

Answering [Q2] is not easy since the inner workings of a typical transformer model are widely distributed across different layers and attention heads with millions of parameters. Large transformer LMs are not only hard to interpret but also tend to underestimate processing difficulties (Oh and Schuler, 2022; Hu et al., 2025). Therefore, we design a small-scale transformer whose internals are easy to interpret and to supervise.

Given this idea, we propose to use a language model that utilises syntactic structure explicitly for its next token prediction mechanism. More concretely, we use a 2-head transformer-based model and train one of its heads to attend to the syntactic governor of the input token whenever it is accessible (i.e. to the left) and the other to attend to its syntactic dependents when they are accessible. This implements a form of incremental parsing. Now we measure structural integration costs based on our model, and show that the joint predictive power of structural cost and surprisal with respect to reading times is significantly larger than the one of only surprisal (from the same model). Finally, we compare the predictive power that the two measures from our syntax-enhanced model together provide with the predictive power that surprisal from a language modelling-only variant of the architecture yields. We establish that our syntax-informed model captures human sentence processing on self-paced reading times better and on eye-tracking data worse than a vanilla model.

We make our code publicly available.[1]

3 Related work

It is well known that reading times correlate with surprisal (Shain et al., 2024). But besides frequency-based theories there are also memory-based theories like Dependency Locality Theory (Gibson, 2000) that establish the contribution of structural effects on reading times. In this paper, we are interested in predicting these effects in reading times. Structural effects can be predicted using syntactic language model parsers (Hale, 2001; Roark, 2001; Hale et al., 2018). Here we take advantage of a relation between attention matrices used in transformer models and attention matrices used in graph-based parsers (Dozat and Manning, 2016) to propose an integration of graph-based parsing into a language model for which we can explicitly add a supervisable structural bias. By doing this, we are close to the recent proposal of Timkey and Linzen (2023) who explored the use of small-sized transformer language models that remain easy to interpret. Our implementation can be seen as a

[1] https://github.com/filemon11/MITransformer

stricter version of their retrieval-based approach where the number of previous tokens to retrieve is minimised and queries/keys are implicitly conditioned to encode syntactic governors/dependents.

Recent work aims to bridge expectation-based and memory-based accounts of language processing by proposing unified models that constrain contextual representations used in prediction. Notably, Futrell et al. (2020) and Hahn et al. (2022) develop frameworks that formalise the trade-offs between memory limitations and predictive efficiency, while Kuribayashi et al. (2022) show that minimising transformer context access generally improves RT predictions. Yet, they also find that for specific syntactic constructions, not strictly determined by dependency length, longer contexts are necessary. They suggest including syntactic biases into context access - a direction our work addresses.

This work also features a form of multitask learning. Collobert and Weston (2008) pioneered the inclusion of several objectives into neural NLP models to improve generalisation and efficiency. More recently, LM architectures like the transformer have been adapted, with approaches such as MT-DNN (Liu et al., 2019) that combine a shared encoder with task-specific output layers. Compared to those approaches, where the precise effect on a model's internal representations remains unclear, our parsing objective has an easily interpretable effect, in that it directly induces patterns in the attention weights of a transformer.

4 Can we observe (structural) effects in reading time data?

First, we investigate the interplay of expectation-based and memory-based theories with respect to human reading times in natural data. In general, it is unclear how they relate to each other. It is possible that tokens with higher integration costs and long-range dependencies are generally rarer, and thus naturally more surprising to the reader. Indeed, Demberg and Keller (2008) find evidence for effects driven by integration costs only for nouns. Thus, we need to establish to what degree RT phenomena are exclusively explicable by costs incurred through memory effects in online processing and not by predictive effects to be able to reasonably judge the contribution of our joint model.

Therefore, we fit linear models to predict reading times from surprisal and dependency-based costs calculated on silver parses. We observe that both theories' contributions are significant and that including memory-based costs in a model that contains surprisal as a predictor significantly improves model fit. This leads us to believe that finding candidates for memory-based metrics in neural language models might allow us to build models that better reflect human processing behaviour.

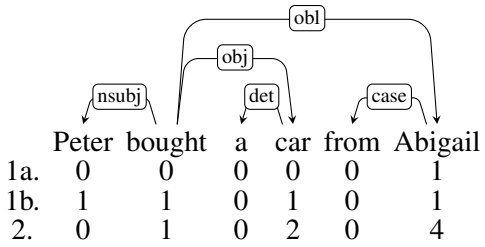


Figure 1: An example dependency parse. Structural integration costs are obtained by summing the linking costs in row 1a. and the establishment costs in row 1b. Costs assigned by leftmost connection distance (LCD) can be found in row 2.

4.1 Data

We utilise the University College London (UCL) corpus of sentences from English narrative sources that comes with both self-paced reading times and eye-tracking data (Frank et al., 2013). It features 361 sentences with an average length of 13.7 words. Self-paced reading times (SPR) were provided by 117 participants while eye-tracking measures were collected from 43 subjects. Regarding the eye-tracking data, we use first fixation duration (FFD), which is the duration of the very first fixation on a word, gaze duration (GD), the summed duration of all fixations on a word before the fixation of any other word, and go-past time (GPT), being the total time spent from first entering a word until moving past it to the right, including any regressions back to earlier text. We include FFD, GD and GPT because we expect to attribute differences in our metrics' ability to predict these measures to regressions or to re-fixations.

For training our model, we generate silver dependency parses using the state-of-the-art spaCy English transformer pipeline.[2]

4.2 Method

Our main predictors are surprisal and structural integration cost. We calculate surprisal using a small-scale LM consisting of an LSTM and a transformer layer with two heads (see Section 5.1 for

[2] https://spacy.io/models/en#en_core_web_trf

more information about the model). We calculate structural integration costs in the following way, similar to Demberg and Keller (2008) and close to the formulation by Gibson (2000): For a given content word (noun or verb), we compute the number of intervening content words between it and its leftmost preceding governor/dependent that is also a content word (0 if none is available), and we add an establishment cost of 1. Non-content words receive a cost of 0. See Figure 1 for an example.

Additionally, we test a modified version of structural integration cost which we call *leftmost connection distance* (LCD). This metric does not ignore non-content words. For each token, it simply yields the distance to its leftmost governor/dependent. If no governor/dependent to the left exists, LCD is 0. This is motivated by Demberg and Keller (2008)'s suggestion that there might be structural phenomena for words where DLT does not predict a cost. Additionally, in contrast to the canonical structural cost, this metric is directly extractable from self-attention matrices as we will discuss in Section 5.3. Figure 1 also contains an example for LCD.

We investigate the correlation between these predictors and human reading times using linear mixed-effects models. Word frequency and word length are included as baseline predictors and random intercepts are included for the participants.

Since processing slowdown is often delayed in RT data (Ehrlich and Rayner, 1983), we add shifted versions of our predictors. For a given S, the amount of spillover, for each word, we not only use the values assigned to this word, but also those of the S previous ones. We decide on S by first fitting a control model to the data without spillover, and then fitting a second model using the same predictors plus a shifted version of the variables. If the latter is a significantly better fit to the data, we choose it, otherwise we stick with the control model. As long as we get significant improvements, we repeat this procedure – up to $S = 2$ in order to avoid losing too much data. Since it generally turns out to be best, we report results for $S = 2$, except when noted otherwise. The test model uniquely adds the metric of interest (e.g. surprisal) and its spillover versions to the baseline.[3]

[3]We will report the following codes: *** highly significant, ** very significant, * significant, . marginally significant. Furthermore, we provide the coefficient estimate (detailed results in Appendix C), ΔLogLik, i.e. the change in log-likelihood after adding the predictor of interest to the model (higher = better) and ΔAIC, i.e. the Akaike Information Criterion (lower = better). The latter two are averaged by the number of observations included (50568).

	coef	ΔLogLik	ΔAIC	p-value
standard surprisal				
SPR	0.22	1.38e-5	-4.10e-6	.
FFD	2.00	1.03e-3	-1.94e-3	***
GD	2.85	1.03e-3	-1.95e-3	***
GPT	3.32	1.31e-3	-2.50e-3	***
GPT2 surprisal				
SPR	0.30	1.71e-4	-3.18e-4	***
FFD	1.34	1.47e-3	-2.82e-3	***
GD	2.13	1.40e-3	-2.67e-3	***
GPT	3.41	2.01e-3	-3.91e-3	***
structural				
SPR	-0.14	1.28e-5	-2.13e-6	.
FFD	0.94	2.32e-4	-3.45e-4	***
GD	1.30	2.12e-4	-3.05e-4	***
GPT	0.07	1.56e-4	-1.93e-4	**
leftmost connection distance				
SPR	0.60	5.22e-5	-8.09e-5	***
FFD	-2.03	4.76e-4	-8.33e-4	***
GD	-1.93	3.45e-4	-5.71e-4	***
GPT	-3.42	6.21e-4	-1.12e-3	***

Table 1: Improvements in mixed linear effects model fit when including one of four predictors: surprisal from our small LM, GPT2 surprisal, structural cost computed on silver parses and LCD computed on silver parses.

4.3 Results

Our results for the predictive power of surprisal, structural integration cost and LCD can be found in Table 1. For comparison with previous research, we also included results for surprisal from the smallest GPT2 model (Radford et al., 2019). We can see that the contribution of surprisal from our baseline model is highly significant for all of the eye-tracking measurements but only marginally significant for self-paced reading times. Self-paced reading times might be noisier and more strategic since participants cannot return to previous material, which can wash out some of surprisal's predictive power. As expected considering the small size of our model, it performs worse than GPT2, with the difference being most notable for self-paced reading times.

Structural integration shows the same pattern. We expected to see more significant results for GPT than for FFD because it includes regressions to the left which we thought to correspond to integration of preceding material. However, the result is contrary, which might be caused by integration cost being entangled with early lexical access or lexical expectations which are believed to manifest more strongly in FFD than in GPT (Conklin et al., 2018). We did not make a hypothesis about GD because it was included post-hoc in response to a review.

Previous research has found a facilitative effect at long dependencies, (among others Konieczny and Döring, 2003; Demberg and Keller, 2008; Rathi, 2021), questioning the explanations provided by DLT. However, we find positive effects for eye-tracking and a small negative effect for SPR times. The coefficient seems to decrease when regressions to previous elements are included (1.30 for GD vs. 0.07 for GPT). Possibly, correlation with surprisal acts as a confounder. However, while further investigations showed a high Pearson correlation of 0.4 between surprisal and structural integration, correlations between FFD/GD and surprisal are only marginally higher than for GPT and surprisal (see Table 8 in Appendix C).

Results for LCD are highly significant for all four dependent variables with ΔAIC ranging from -8.09e-5 for SPR to -1.12e-3 for GPT. This is noticeably better than the canonical structural cost and might indicate that non-content words influence memory-based costs both in terms of calculating the distance function and as cost-carrying words themselves. It is also possible that the class of content words should contain additional categories of words that we left out, e.g. adjectives.

For the coefficient, here we find inverted results with the sign being negative for eye-tracking and positive for SPR. Interestingly, higher coefficients for surprisal seem to coincide with lower coefficients for LCD. Possibly eye movements reflect a more shallow form of *good-enough processing* (Ferreira et al., 2002), as suggested by Kuribayashi et al. (2022), more strongly influenced by frequency effects, while SPR might be more strategic as noted above, due to the inaccessibility of preceding information and more influenced by structural integration. The stronger anti-locality effect for LCD where surprisal is most predictive would then be explicable by a frequency-based account, i.e. the accumulation of probabilistic evidence, for instance, before clause final verbs (Levy, 2008).

Due to the more significant results LCD provides and our ability to extract it from our LM, we stick with it for the remainder of the paper.

Naturally, the question arises of how LCD and surprisal behave with respect to each other and whether we can disentangle their effects. The Pearson correlation between surprisal and this measure is lower than for structural integration cost (0.19), so it seems less likely that we observe frequency effects. This may also be partly explained by the fact that in contrast to structural cost, LCD takes into account non-content words, which generally feature significantly lower surprisal than content words (see Figure 6 in the Appendix).

spill		coef	ΔLogLik	ΔAIC	p-value
leftmost connection distance over standard surprisal					
0	SPR	0.54	3.64e-5	-6.61e-5	***
	FFD	-2.27	2.37e-4	-4.44e-4	***
	GD	-4.52	7.20e-4	-1.41e-3	***
	GPT	-5.07	6.67e-4	-1.31e-3	***
1	SPR	0.70	7.48e-5	-1.35e-4	***
	FFD	-1.58	3.13e-4	-5.57e-4	***
	GD	-2.55	4.35e-4	-8.02e-4	***
	GPT	-4.23	9.67e-4	-1.85e-3	***
2	SPR	0.62	5.46e-5	-8.58e-5	***
	FFD	-1.92	4.04e-4	-6.90e-4	***
	GD	-1.79	3.11e-4	-5.03e-4	***
	GPT	-3.19	5.34e-4	-9.49e-4	***

Table 2: Results of including LCD cost in a linear mixed effects model with surprisal as part of the control.

We check whether including LCD in a linear mixed model that contains surprisal as well as our baseline predictors significantly improves the fit. Table 2 shows detailed results, including values for spillover 0, 1 and 2. Again, the selection process established a window of 2 as most relevant.

We can see that structural effects are highly significant across all dependent variables and all spillover window sizes. For SPR and GPT ΔAIC is strongest with -1.35e-4 and -1.85e-3 respectively at spillover 1 while for FFD it is best at spillover 2 with -6.90e-4 and for GD at spillover 0 with -1.41e-3. The trend of a negative sign for SPR and a positive sign for eye-tracking data still holds. Thus, it is unlikely that this phenomenon can be fully explained by a frequency-based account.

These observations suggest that we can answer [Q1] by confirming [H1]: syntactic integration costs impact processing in a measurable and sustained way that is not fully captured by surprisal.

5 Can a syntax-informed model better capture human processing?

5.1 Models

In order to address question [Q2] of whether a syntax-informed language model better captures features of human sentence processing, we design two small-scale language models. The first model (called *standard*) serves as our baseline and is trained for next token prediction while the second model (called *supervised*) receives an additional incremental dependency parsing objective. More concretely, in our syntax-enhanced model, dependency edges are represented via the attention each

token pays to the items that precede it. To this end, we train one attention head so that each token attends to its governor if its on the left, and one attention head so that each token attends to all of its dependents on the left. The model is trained in a multitask fashion, where the loss is a weighted average of a language modelling loss $Loss_{LM}$ (cross-entropy) and two syntactic losses $Loss_{syn} = (Loss_{gov} + Loss_{dep})/2$ (binary cross-entropy) given in Equation 1.

$$Loss = \alpha\ Loss_{LM} + (1 - \alpha)\ Loss_{syn} \quad (1)$$

The optimal weight for language modelling and parsing is non-trivial to select and is therefore determined through hyperparameter optimisation, as are learning rate, dimensionality and regularisation strengths. For the standard model, we select hyperparameters that minimise perplexity and for the supervised model, we select hyperparameters that maximise *unlabelled attachment score* (UAS), that is, the percentage of tokens assigned the correct governor. For the loss-term α in the supervised setting, the search yields an optimal value of 0.05 which is heavily leaning towards parsing. Additional information on the hyperparameter search and the resulting parameters can be found in Section B of the appendix. The final models are both trained for 10 epochs.

Our models are based on the transformer architecture. They are causal, meaning that attention heads are constrained to tokens in the left context by masking. See Vaswani et al. (2017) for a detailed introduction to transformers. The schemes for positional encodings and the language modelling head correspond to the GPT architecture (Radford et al., 2018). Following Timkey and Linzen (2023), we contextualise our embeddings using a unidirectional LSTM (Hochreiter and Schmidhuber, 1997) before providing them to the transformer module.

5.2 Data

In the following, we explain our choice of datasets and the pre-processing for training and evaluation.

Training and LM evaluation We use the pre-processed Wikitext-103-v1 dataset[4] for training our models. It consists of over 100 million tokens from Wikipedia. Here, we also use the spaCy trf model to generate silver dependencies, as outlined in 4.1. Before parsing Wikitext and training the model we convert the data to lowercase and apply additional modifications outlined in Section A of the appendix. Finally, we tokenise the dataset on the word-level.

Psycholinguistic evaluation In order to investigate the psycholinguistic plausibility of our models, we again use the UCL corpus (cf. Section 4.1). We treat these sentences as our stimulus and are aware that their domain differs from Wikitext. However, we do not regard this as problematic since we are only interested in comparing the psycholinguistic properties of our models against each other.

5.3 Evaluation

In the next section, we evaluate our models in three respects: (a) language modelling, (b) dependency parsing, and (c) correlation between model measures and human reading times. In the following, we explain our methods of evaluation and how they are used to answer [Q2].

Language modelling We evaluate language modelling capabilities using perplexity.

Dependency parsing The two attention heads together provide a score for each possible dependency in the sentence. We decode these scores as a directed maximum spanning tree using Chu–Liu/Edmonds' algorithm (Chu and Liu, 1965; Edmonds, 1967). Then, we evaluate the prediction by computing UAS. Furthermore, we report the entropy of the probability distributions over preceding tokens provided by the attention heads averaged by all tokens.

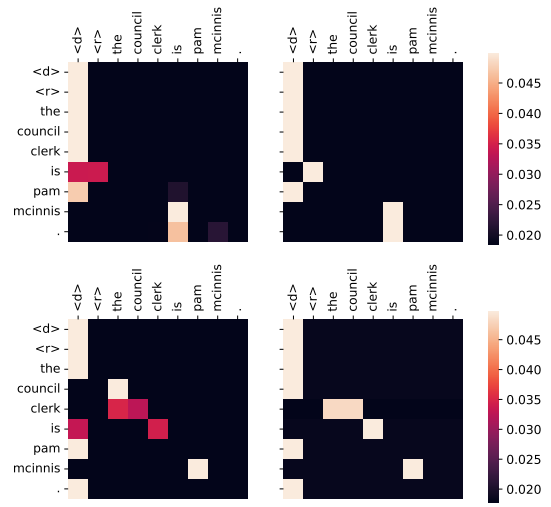
Psycholinguistic plausibility We restrict this analysis exclusively to measures provided by the LM: (i) surprisal and (ii) the attention patterns of our model which we use to compute a prediction for leftmost connection distance (PLCD). This is done by identifying the token with the maximum weight assigned per attention head and then taking the one closest to the beginning of the sentence. If both heads connect to one of two special tokens called "root" and "dummy" (representing a lack of left connections), we manually assign a cost of 0. An example can be found in Figure 2.

5.4 Model comparisons

5.4.1 General performance

Measures of the language modelling performance of our standard and our syntax-enhanced model

[4] https://huggingface.co/datasets/Salesforce/wikitext



	the	council	clerk	is	pam	mcinnis	.
a.	0	1	1	0	0	2	3
b.	0	0	2	1	0	2	3

Figure 2: First row: governor matrix; second row: dependent matrix; first column: prediction; second column: silver adjacency matrices. a. cost predicted by our supervised model; b. cost measured on the silver parse.

model	split	PPL	UAS	attn entropy	
				gov	dep
standard	train	28.46		1.16	1.10
	dev	44.81		1.23	1.15
	test	46.75		1.21	1.15
supervised	train	42.85	0.92	0.06	0.05
	dev	57.13	0.87	0.08	0.06
	test	58.88	0.87	0.08	0.06
CBR-RNN	test	61.8			
GPT2	train	105.00		1.14	
	dev	95.97		1.21	
	test	98.68		1.21	

Table 3: General evaluation of our language models on the Wikitext corpus. PPL = perplexity, UAS = unlabelled attachment score. CBR-RNN (α=0, reported by Timkey and Linzen (2023)) and GPT2 with sentence-based PPL on the raw Wikitext corpus are included for comparison. Note that results for GPT2 are not directly comparable because of the different tokenisation scheme and CBR-RNN neither due to PPL being chunk-based.

on the training, development, and test splits of the Wikitext dataset can be found in Table 3. With a value of 46.75 on the test split, the perplexity of our standard model is considerably lower than the mean perplexity of 61.8 Timkey and Linzen (2023) report for their base model. This is probably due to our model having a dimensionality of 886, while their model comes with a width of just 256.

The perplexity of our supervised model on the test set amounts to 58.88 which is noticeably higher than the standard model, likely caused by the strong focus on dependency parsing (cf. Section 5.1). Structuring the attention mechanism, used to compute the output embeddings of the transformer layer, along dependency arcs might undervalue the role of certain types of preceding context necessary for next token prediction, for instance, when a token should be most probable that is not directly connected with the current item or any of the retrieved directly syntactically connected content. Another possibility might be that reaching good performance for both parsing and language modelling would necessitate a larger model, as hyperparameter optimisation for the supervised model resulted in roughly half the number of parameters than for the standard model (104M vs. 219M). It is also possible that we would have needed more training data and/or longer training to support both tasks. At least the latter is unlikely since we have reached convergence (see Appendix B).

For parsing, the supervised model reaches an UAS of 0.87. Note that this is measured against silver data generated by an off-the-shelf parser – albeit a performant one, with an UAS of 0.95 (Honnibal et al.) on the development set of the OntoNotes 5.0 corpus (Weischedel et al., 2013). Finally, attention is on average much more narrowly distributed in the supervised model (0.08 governor head entropy, 0.06 dependent head) than in the standard model (1.21, 1.15). As an entropy of 0 would correspond to a one-hot vector, this confirms that our training scheme has optimised the model to retrieve information from a minimal number of preceding tokens.

The significance of surprisal and leftmost connection distance extracted from attention patterns of our supervised model (PLCD) for reading time predictions can be found in Table 4. Surprisal significantly improves prediction across all reading time measures, with large gains in FFD, GD and GPT. The benefit for SPR is weaker, but still highly significant. This is noteworthy since the predictive power of surprisal from the standard model was only marginally significant for SPR (cf. Table 1). On the other hand, for the eye tracking measures ΔAIC is lower using standard model surprisal. Overall, despite heavily modifying the attention architecture and yielding an increase in perplexity, surprisal, as a measure of word predictability, is still a strong predictor of reading difficulty.

	coef	ΔLogLik	ΔAIC	p-value
supervised surprisal				
SPR	0.38	4.69e-5	-7.04e-5	***
FFD	2.67	5.16e-4	-9.12e-4	***
GD	3.17	4.93e-4	-8.68e-4	***
GPT	3.78	6.15e-4	-1.11e-3	***
predicted leftmost connection distance				
SPR	0.40	2.01e-5	-1.67e-5	*
FFD	-0.41	1.69e-4	-2.20e-4	**
GD	0.07	1.15e-4	-1.12e-4	**
GPT	-1.26	2.80e-4	-4.42e-4	***

Table 4: Improvements in mixed linear effects model fit when including surprisal or PLCD extracted from our supervised model.

spill		coef	ΔLogLik	ΔAIC	p-value
predicted leftmost connection distance over supervised surprisal					
0	SPR	0.45	2.32e-5	-3.97e-5	***
	FFD	-1.93	1.59e-4	-2.87e-4	***
	GD	-3.78	4.53e-4	-8.77e-4	***
	GPT	-4.75	5.32-4	-1.03e-3	***
1	SPR	0.57	4.92e-5	-8.40e-5	***
	FFD	-1.18	1.67e-4	-2.66e-4	***
	GD	-1.19	1.51e-4	-2.34e-4	***
	GPT	-2.84	5.94e-4	-1.12e-3	***
2	SPR	0.46	2.66e-5	-2.97e-5	**
	FFD	-0.24	1.01e-4	-8.22e-5	*
	GD	0.26	6.79e-5	-1.70e-5	.
	GPT	-0.98	1.69e-4	-2.19e-4	**

Table 5: Results of including PLCD from our supervised model in a linear mixed effects model with surprisal as part of the baseline.

5.4.2 Psycholinguistic performance

The predictive power of PLCD is significant for the four metrics, ranging from -1.67e-5 (SPR) to -4.42e-4 (GPT) ΔAIC. While being less significant than the distance extracted from the silver data as we have reported in Table 1, we have to remind the reader that expectation-based and memory-based effects are entangled in this test, so that greater predictive power of one of the syntactic costs could also be due to correlation with surprisal.

It has to be noted that the estimated coefficients for PLCD on eye-tracking exhibit less than half of the magnitude of the tree-extracted predictor (cf. Table 1). The coefficient for GD even turns out to be positive (0.07). Either this is a result of lower quality syntactic information due to our weaker parsing score or a consequence of the probabilistic, incremental parsing process.

Next, we estimate the improvement that PLCD provides over a model that only includes surprisal as a fixed effect (Table 5). The predicted distance to the leftmost governor/dependent adds significant explanatory power beyond surprisal with all spillover window sizes except for spillover 2 and GD. For SPR, ΔAIC is lowest for spillover 1 while for FFD, GD and GPT it is lowest for spillover 0 and increases strongly at window size 2, still yielding significant/very significant results. Thus, the predictive power of memory cost decreases when preceding surprisals (and other predictors) are included. Overall, results for spillover 2 are significant for most measures and using both surprisal and PLCD should improve reading time predictions.

Finally, to answer [Q2], we determine the predictive power of surprisal and memory-based costs compared to the linear mixed-effects control model. Results can be found in Table 6. Combining surprisal and PLCD from our supervised model beats standard model surprisal paired with structural cost from silver parses for self-paced reading times slightly (ΔAIC -1.00e-4 vs. ΔAIC -8.99e-5) but yields roughly a third of the ΔAIC for eye-tracking measurements. All results are highly significant.

Comparing with the predictions we could extract from the standard model (only surprisal, cf. Table 1), we can establish that we achieved highly significant results for self-paced reading times where standard surprisal was only marginally significant. However, for eye-tracking, the fit is better using surprisal from the unrestricted model. Therefore, we can confirm [H2] in part: A syntax-informed model seems to better reflect human processing for self-paced reading data than surprisal from a vanilla language model, whereas the opposite is true for eye-tracking data.

6 Conclusion

Summary The contribution of this paper is twofold. First, we have shown that RT predictions significantly improve when considering not only surprisal (obtained from a standard generative LM), i.e. expectation-based measures, but also structural integration costs obtained from parse trees using an off-the-shelf parser. This confirms insights from the psycholinguistic literature (e.g. Gibson, 2000) at a larger scale, i.e. on a corpus annotated with reading times. However, the direction of the effect seems to depend on the type of measurements.

Building on this, our second contribution consists of a proposal for a syntax-enhanced generative LM that produces not only next word predictions (and thereby surprisal) but also predictions of dependency edges to the left, which can serve to

	coef1	coef2	ΔLogLik	ΔAIC	p-value
standard surprisal + leftmost connection distance					
SPR	0.21	0.62	6.84e-5	-8.99e-5	***
FFD	2.16	-1.92	1.43e-3	-2.63e-3	***
GD	3.02	-1.79	1.34e-3	-2.45e-3	***
GPT	3.51	-3.19	1.84e-3	-3.45e-3	***
supervised surprisal + predicted leftmost connection distance					
SPR	0.41	0.46	7.35e-5	-1.00e-4	***
FFD	2.57	-0.24	6.16e-4	-9.94e-4	***
GD	3.12	0.26	5.61e-4	-8.85e-4	***
GPT	3.57	-0.98	7.84e-4	-1.33e-3	***

Table 6: Effect of including both (P)LCD and (supervised) surprisal in a linear mixed effects model.

compute syntactic integration costs. Even though the quality of the parse trees is below that of the off-the-shelf parser (partly because of the strict incrementality of the parser), the additional structural predictions, when quantified as integration costs, increase the predictive power of the model concerning reading times compared to using just surprisal values from the same model. In other words, we implemented an incremental model that yields expectation-based and memory-based RT predictors, similar to what we observed as relevant in the experiments for our first contribution.

Discussion We have found that the RT measurements we used are quite different in nature: In regards to eye-tracking, we could observe that the predictive power of (predicted) leftmost connection distance is higher for GPT than for FFD and GD, throughout the experiments. This indicates that part of the memory-based processing effect might express itself through regressions to preceding words. For SPR, the role of memory effects is harder to analyse, which might have to do with the stronger level of spillover effects generally found in this paradigm (Frank et al., 2013; Witzel et al., 2012), leading to a diffuse distribution of expectation-based and memory-based costs.

For eye-tracking, our joint predictive model falls short of the improvements provided by standard surprisal and syntactic cost extracted from silver parses. We think that this is due to the fact that surprisal from the syntax-enhanced model alone already exhibited a worse fit to the data than surprisal from the vanilla model. Thus, the contributions of integration cost could not compensate for the lower baseline. Here, we see potential in designing an architecture with better language modelling capabilities while maintaining the syntactic objective.

As to the estimated coefficients of the memory-effect, our results are mixed. The finding of anti-locality effects for eye-tracking is in agreement with previous research (e.g. Konieczny and Döring, 2003; Demberg and Keller, 2008; Rathi, 2021). However, the fact that we can still see significant anti-locality contributions even if we include surprisal does not point towards a frequency-based explanation. Possibly, our observations support the theory of dynamic recruitment of additional processing resources, as proposed by Just and Varma (2007), where increased costs occur at the start of embedded constructions due to the activation of additional cognitive resources and facilitation occurs at the end, where the reader still has temporary access to those capacities. Assuming SPR to reflect a more strategic processing, it might be possible that these resources are in a state of more constant activation, so that anti-locality effects cannot be observed. In the end, the question would remain whether the positive coefficient for SPR hints to true locality effects as predicted by DLT.

Concerning the dependency enhanced LM, as mentioned, strict (left-to-right) incrementality decreases parsing accuracy. When it comes to predicting human processing, this is probably an advantage. Compared to structural costs derived from gold or near-gold parses, incrementally predicted structural costs can be expected to be more predictive of reading times since they probably reflect uncertainty of the parser in situations that can only be disambiguated through right context. However, we do not claim that the parser implemented in this paper is cognitively plausible. It has been argued (for instance by Demberg et al., 2013) that for a parser to be psycholinguistically plausible, the parser not only has to be incremental but also predictive (i.e. predicting upcoming words and structure) and connected (i.e. the syntactic contribution of a new word has to be immediately integrated into the already built prefix tree). However, our dependency enhanced LM does not predict a connected graph at each step. (For parsing accuracy evaluation, a tree is constructed in a post-processing step.) Furthermore, while our model predicts the next word, it does not make any prediction about the upcoming structure.

7 Acknowledgements

We would like to thank the three anonymous reviewers for their valuable and helpful feedback.

Parts of this study were done during an Eras-

mus+ traineeship at the Laboratoire de linguistique formelle at Université Paris Cité.

References

Yoeng-Jin Chu and Tseng-Hong Liu. 1965. On the shortest arborescence of a directed graph. In *Science Sinica*, volume 14, pages 1396–1400.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA. Association for Computing Machinery.

Kathy Conklin, Ana Pellicer-Sanchez, and Gareth Carrol. 2018. *Eye-Tracking: A Guide for Applied Linguistics Research*. Cambridge University Press.

Vera Demberg and Frank Keller. 2008. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109(2):193–210.

Vera Demberg, Frank Keller, and Alexander Koller. 2013. Incremental, predictive parsing with psycholinguistically motivated Tree-Adjoining Grammar. *Computational Linguistics*, 39(4):1025–1066.

Timothy Dozat and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing. *CoRR*, abs/1611.01734.

Jack Edmonds. 1967. Optimum branchings. In *Journal of Re- search of the National Bureau of Standards*, volume 71B, pages 233–240.

Kate Ehrlich and Keith Rayner. 1983. Pronoun assignment and semantic integration during reading: eye movements and immediacy of processing. *Journal of Verbal Learning and Verbal Behavior*, 22(1):75–87.

Ana Ezquerro, Carlos Gómez-Rodríguez, and David Vilares. 2024. From partial to strictly incremental constituent parsing. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 225–233, St. Julian's, Malta. Association for Computational Linguistics.

Fernanda Ferreira, Karl G.D. Bailey, and Vittoria Ferraro. 2002. Good-Enough Representations in Language Comprehension. *Current Directions in Psychological Science*, 11(1):11–15.

Stefan L. Frank, Irene Fernandez Monsalve, Robin L. Thompson, and Gabrielle Vigliocco. 2013. Reading time data for evaluating broad-coverage models of English sentence processing. *Behavior Research Methods*, 45(4):1182–1190.

Richard Futrell, Edward Gibson, and Roger P. Levy. 2020. Lossy-Context Surprisal: An Information-Theoretic Model of Memory Effects in Sentence Processing. *Cognitive Science*, 44(3):e12814.

Edward Gibson. 2000. The dependency locality theory: A distance-based theory of linguistic complexity. In *Image, Language, Brain: Papers from the First Mind Articulation Project Symposium*.

Michael Hahn, Richard Futrell, Roger Levy, and Edward Gibson. 2022. A resource-rational model of human processing of recursive linguistic structure. *Proceedings of the National Academy of Sciences of the United States of America*, 119(43):e2122602119.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

John Hale, Chris Dyer, Adhiguna Kuncoro, and Jonathan Brennan. 2018. Finding syntax in human encephalography with beam search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2727–2736, Melbourne, Australia. Association for Computational Linguistics.

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

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy Models Facts & Figures. Accessed on: 2025-08-10.

Jennifer Hu, Michael A. Lepori, and Michael Franke. 2025. Signatures of human-like processing in transformer forward passes.

Marcel Adam Just and Sashank Varma. 2007. The organization of thinking: What functional brain imaging reveals about the neuroarchitecture of complex cognition. *Cognitive, Affective, & Behavioral Neuroscience*, 7(3):153–191.

Lars Konieczny and Philipp Döring. 2003. Anticipation of clause-final heads: Evidence from eye-tracking and SRNs. In *Proceedings of iccs/ascs*, pages 13–17. Sydney, NSW.

Tatsuki Kuribayashi, Yohei Oseki, Ana Brassard, and Kentaro Inui. 2022. Context Limitations Make Neural Language Models More Human-Like. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10421–10436. Association for Computational Linguistics.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106:1126–1177.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

Byung-Doh Oh and William Schuler. 2022. Why does surprisal from larger transformer-based language models provide a poorer fit to human reading times?

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAi Blog*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAi Blog*.

Neil Rathi. 2021. Dependency Locality and Neural Surprisal as Predictors of Processing Difficulty: Evidence from Reading Times. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 171–176. Association for Computational Linguistics.

Brian Roark. 2001. Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 27(2):249–276.

Soo Hyun Ryu and Richard Lewis. 2021. Accounting for agreement phenomena in sentence comprehension with transformer language models: Effects of similarity-based interference on surprisal and attention. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 61–71, Online. Association for Computational Linguistics.

Cory Shain, Clara Meister, Tiago Pimentel, Ryan Cotterell, and Roger Levy. 2024. Large-scale evidence for logarithmic effects of word predictability on reading time. *Proceedings of the National Academy of Sciences*, 121(10):e2307876121.

William Timkey and Tal Linzen. 2023. A language model with limited memory capacity captures interference in human sentence processing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8705–8720, Singapore. Association for Computational Linguistics.

Julie A Van Dyke and Richard L Lewis. 2003. Distinguishing effects of structure and decay on attachment and repair: A cue-based parsing account of recovery from misanalyzed ambiguities. *Journal of Memory and Language*, 49(3):285–316.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Ann Taylor Lance Ramshaw, Nianwen Xue, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, and Ann Houston Robert Belvin. 2013. OntoNotes release 5.0 LDC2013T19. Linguistic Data Consortium. Accessed on: 2025-08-21.

Ethan G. Wilcox, Tiago Pimentel, Clara Meister, Ryan Cotterell, and Roger P. Levy. 2023. Testing the predictions of surprisal theory in 11 languages. *Transactions of the Association for Computational Linguistics*, 11:1451–1470.

Naoko Witzel, Jeffrey Witzel, and Kenneth Forster. 2012. Comparisons of online reading paradigms: Eye tracking, moving-window, and maze. *Journal of Psycholinguistic Research*, 41:105–128.

Kaiyu Yang and Jia Deng. 2020. Strongly incremental constituency parsing with graph neural networks. *CoRR*, abs/2010.14568.

setting	standard	supervised
goal	PPL	UAS
trials	65	65
optimal value	43.68	0.88
drop resid	0.219	0.597
drop ff	0.026	0.454
drop embd	0.083	0.133
drop lstm	0.305	0.211
n embd	886	464
d ff factor	7	4
alpha		0.05
lr	1.21e-3	4.13e-4
parameters	219,113,116	104,334,112

Table 7: Results of hyperparameter optimisation for the standard and for the supervised model.

Appendices

A Data preprocessing

For our neural language models, we preprocess the datasets in the following way:

1. Lowercase the text.
2. Remove titles (starting with "=").
3. Remove lines with more than 4 white space-separated tokens.
4. Replace "@-@" with "-", " @,@ " with "," and " @.@ " with ".". These symbols were artificially introduced into the Wikitext corpus.
5. Replace numbers by <num>. The heuristic is checking if a token consists only of numerals after removing all dots, commas and hyphens in it.

Furthermore, we remove all sentences with less than 5 words and, additionally for training, all sentences with more than 40 words.

B Optimisation and training

Hyperparameter optimisation We perform hyperparameter optimisation separately for the standard model and for the supervised model. The results of this process can be found in Table 7. We started the optimisation with seed 1895 and incremented it for each training round. Note that we round the optimal hyperparameters in Table 7. We also used these rounded values for the full training. For the full training, we use seed 1895.

Computational resources Training was performed using four H100 GPUs with a batch size of 512 on the RWTH Aachen CLAIX cluster.

	FFD	GD	GPT	struct	surpr
GD	0.90				
GPT	0.59	0.63			
struct	0.16	0.17	0.11		
surpr	0.22	0.24	0.16	0.40	
LCD	0.06	0.07	0.04	0.61	0.19

Table 8: Correlations between FFD, GD, GPT, structural integration cost, surprisal from our standard model and LCD.

	SPR	struct	surpr
struct	0.00		
surpr	0.02	0.35	
LCD	0.00	0.66	0.16

Table 9: Correlations between SPR, structural integration cost, surprisal from our standard model and LCD.

Training You can find plots of language modelling loss on the Wikitext train and development split during the training of our supervised model in Figure 3. Language modelling loss and parsing loss during the training of our supervised model is given in Figures 4 and 5. We chose the model snapshot with the best performance on the validation split.

C Psycholinguistic evaluation

We include correlations between the metrics used in Experiment 1 as well as a plot showing the average surprisal per POS tag (Tables 8, 9 and Figure 6). Furthermore, we include detailed results including all estimated coefficients for the linear mixed-effect models fitted in our experiments in Tables 10 to 21.

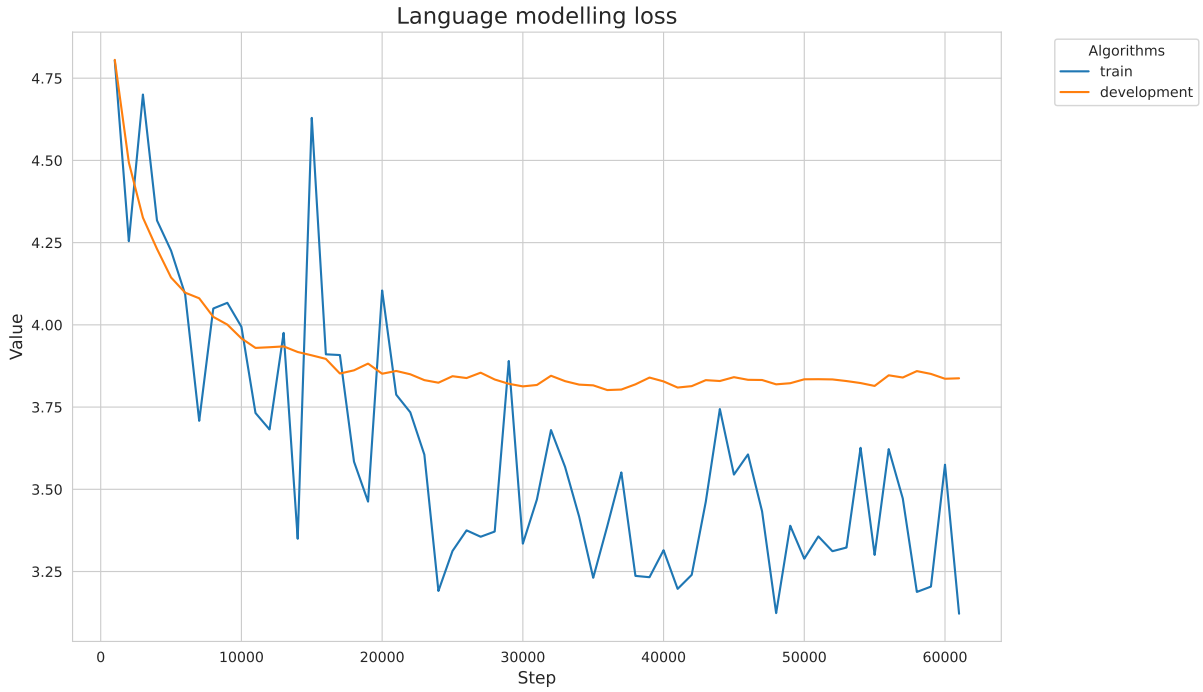


Figure 3: Development of language modelling loss on the train and on the development set during the training process of the standard model. The rightmost value corresponds to epoch 10.

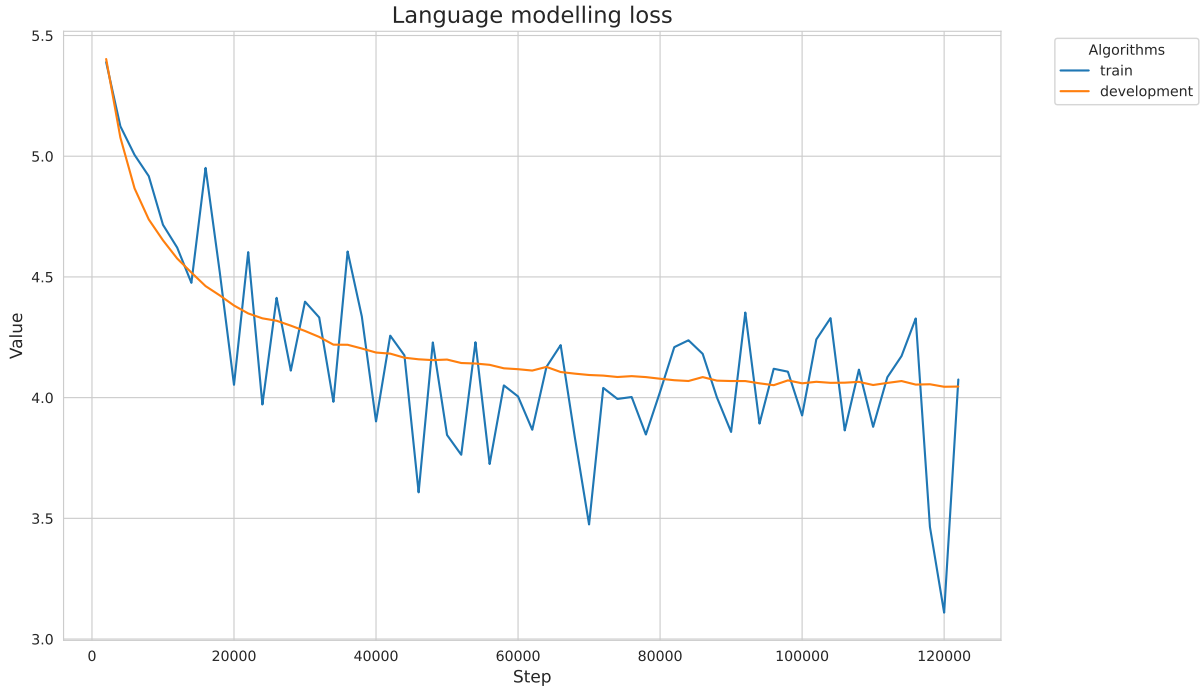


Figure 4: Development of language modelling loss on the train and on the development set during the training process of the supervised model. The rightmost value corresponds to epoch 10.

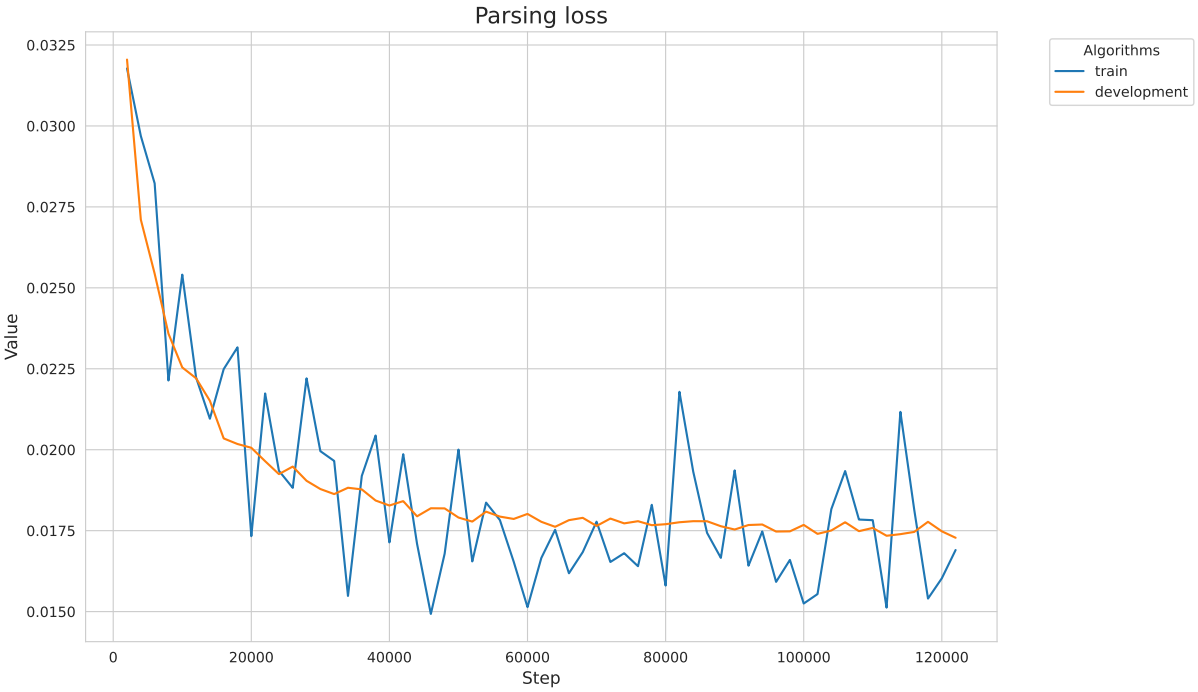


Figure 5: Development of parsing loss on the train and on the development set during the training process of the supervised model. The rightmost value corresponds to epoch 10.

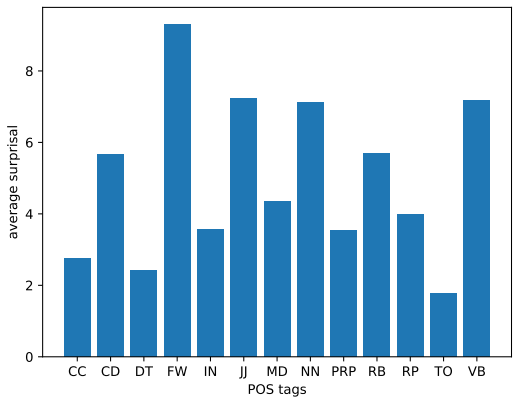


Figure 6: Average surprisal of our standard model by POS tags. We use the POS tags provided by the spaCy pipeline and reduce the number of distinct sets by merging.

	effect	coef	Std. Error	t-value
standard surprisal				
SPR	intercept	272.15	5.23	52.02
	frequency_0	0.06	0.21	0.28
	frequency_1	-2.10	0.22	-9.76
	frequency_2	-0.95	0.21	-4.55
	length_0	0.64	0.18	3.58
	length_1	1.08	0.18	6.00
	length_2	0.62	0.18	3.52
	surprisal_0	0.22	0.17	1.26
	surprisal_1	0.36	0.18	2.07
	surprisal_2	-0.00	0.17	-0.02
FFD	intercept	117.78	4.18	28.19
	frequency_0	-11.79	0.73	-16.15
	frequency_1	3.23	0.74	4.34
	frequency_2	-1.80	0.72	-2.50
	length_0	25.83	0.62	41.53
	length_1	-9.43	0.62	-15.23
	length_2	1.71	0.61	2.78
	surprisal_0	2.00	0.61	3.27
	surprisal_1	5.40	0.62	8.75
	surprisal_2	-1.18	0.60	-1.94
GD	intercept	128.44	5.00	25.67
	frequency_0	-13.31	0.81	-16.43
	frequency_1	2.87	0.83	3.48
	frequency_2	-2.67	0.80	-3.34
	length_0	31.88	0.69	46.16
	length_1	-11.39	0.69	-16.57
	length_2	1.97	0.68	2.90
	surprisal_0	2.85	0.68	4.21
	surprisal_1	5.63	0.69	8.20
	surprisal_2	-1.47	0.67	-2.18
GPT	intercept	139.36	5.01	27.82
	frequency_0	-13.15	0.94	-14.04
	frequency_1	4.08	0.95	4.27
	frequency_2	-2.82	0.92	-3.05
	length_0	34.23	0.80	42.90
	length_1	-13.71	0.79	-17.26
	length_2	0.36	0.78	0.46
	surprisal_0	3.32	0.78	4.23
	surprisal_1	7.52	0.79	9.49
	surprisal_2	-0.75	0.78	-0.97

Table 10: Detailed results for fitting a mixed linear effects model including surprisal from a small vanilla LM as a fixed effect, as well as our baseline predictors. Shifted predictors for spillover window sizes 1 and 2 are also included.

	effect	coef	Std. Error	t-value
structural				
SPR	intercept	272.15	5.23	52.02
	frequency_0	-0.14	0.19	-0.72
	frequency_1	-2.43	0.20	-12.9
	frequency_2	-0.99	0.19	-5.18
	length_0	0.66	0.18	3.71
	length_1	1.15	0.18	6.43
	length_2	0.66	0.18	3.74
	structural_0	-0.14	0.14	-1.00
	structural_1	-0.30	0.14	-2.14
	structural_2	-0.16	0.14	-1.15
FFD	intercept	117.79	4.18	28.20
	frequency_0	-12.77	0.70	-18.36
	frequency_1	-1.40	0.70	-2.01
	frequency_2	-0.70	0.68	-1.02
	length_0	26.02	0.62	41.97
	length_1	-8.65	0.61	-14.11
	length_2	1.88	0.61	3.08
	structural_0	0.94	0.52	1.79
	structural_1	-2.17	0.52	-4.21
	structural_2	0.66	0.51	1.29
GD	intercept	128.45	5.00	25.68
	frequency_0	-14.64	0.77	-18.95
	frequency_1	-1.85	0.77	-2.40
	frequency_2	-1.27	0.76	-1.67
	length_0	32.21	0.69	46.79
	length_1	-10.50	0.68	-15.42
	length_2	2.08	0.68	3.07
	structural_0	1.30	0.58	2.23
	structural_1	-2.00	0.57	-3.50
	structural_2	1.02	0.57	1.79
GPT	intercept	139.36	5.01	27.84
	frequency_0	-15.57	0.89	-17.44
	frequency_1	-1.89	0.89	-2.12
	frequency_2	-1.66	0.88	-1.89
	length_0	34.58	0.80	43.47
	length_1	-12.47	0.79	-15.85
	length_2	0.73	0.78	0.94
	structural_0	0.07	0.67	0.11
	structural_1	-2.32	0.66	-3.51
	structural_2	1.18	0.66	1.79

Table 11: Detailed results for fitting a mixed linear effects model including structural integration cost as a fixed effect, as well as our baseline predictors. Shifted predictors for spillover window sizes 1 and 2 are also included.

	effect	coef	Std. Error	t-value
leftmost connection distance				
SPR	intercept	272.15	5.23	52.02
	frequency_0	0.03	0.19	0.16
	frequency_1	-2.36	0.19	-12.55
	frequency_2	-1.00	0.18	-5.54
	length_0	0.66	0.18	3.67
	length_1	1.12	0.18	6.21
	length_2	0.56	0.18	3.15
	LCD_0	0.60	0.13	4.78
	LCD_1	0.08	0.13	0.67
	LCD_2	0.26	0.13	2.09
FFD	intercept	117.79	4.18	28.20
	frequency_0	-13.61	0.65	-20.97
	frequency_1	-0.62	0.65	-0.95
	frequency_2	-0.86	0.62	-1.39
	length_0	25.99	0.62	41.83
	length_1	-8.28	0.61	-13.52
	length_2	1.98	0.61	3.25
	LCD_0	-2.03	0.45	-4.54
	LCD_1	-2.13	0.45	-4.73
	LCD_2	1.16	0.45	2.55
GD	intercept	128.45	5.00	25.68
	frequency_0	-15.74	0.72	-21.83
	frequency_1	-1.11	0.72	-1.55
	frequency_2	-1.59	0.69	-2.31
	length_0	32.28	0.69	46.77
	length_1	-10.08	0.68	-14.83
	length_2	2.09	0.68	3.08
	LCD_0	-1.93	0.50	-3.88
	LCD_1	-1.43	0.50	-2.86
	LCD_2	1.79	0.50	3.55
GPT	intercept	139.36	5.01	27.83
	frequency_0	-16.24	0.83	-19.50
	frequency_1	-1.28	0.83	-1.54
	frequency_2	-2.03	0.79	-2.56
	length_0	34.57	0.80	43.35
	length_1	-12.04	0.79	-15.33
	length_2	0.81	0.78	1.04
	LCD_0	-3.42	0.57	-5.95
	LCD_1	-2.64	0.58	-4.57
	LCD_2	1.64	0.58	2.82

Table 12: Detailed results for fitting a mixed linear effects model including leftmost connection distance as a fixed effect, as well as our baseline predictors. Shifted predictors for spillover window sizes 1 and 2 are also included.

spill		effect	coef	Std. Error	t-value
leftmost connection distance and standard surprisal					
0	SPR	intercept	272.35	5.25	51.87
		frequency_0	0.91	0.19	4.66
		length_0	0.36	0.17	2.16
		surprisal_0	0.77	0.16	4.83
		LCD_0	0.54	0.12	4.69
	FFD	intercept	122.29	4.74	25.78
		frequency_0	-13.09	0.67	-19.32
		length_0	26.91	0.57	47.42
		surprisal_0	0.47	0.56	0.83
		LCD_0	-2.27	0.40	-5.64
	GD	intercept	137.17	5.73	23.94
		frequency_0	-15.54	0.78	-20.04
		length_0	34.84	0.65	53.64
		surprisal_0	3.23	0.64	5.03
		LCD_0	-4.52	0.46	-9.83
	GPT	intercept	150.56	5.88	25.62
		frequency_0	-16.57	0.90	-18.37
		length_0	38.38	0.76	50.80
		surprisal_0	3.56	0.75	4.78
		LCD_0	-5.07	0.54	-9.47

Table 13: Detailed results for fitting a mixed linear effects model including surprisal from a small vanilla LM and leftmost connection distance as fixed effects, as well as our baseline predictors, without any spillover.

spill		effect	coef	Std. Error	t-value
leftmost connection distance and standard surprisal					
1	SPR	intercept	271.92	5.25	51.75
		frequency_0	0.24	0.20	1.15
		frequency_1	-1.58	0.20	-7.80
		length_0	0.54	0.17	3.14
		length_1	1.14	0.17	6.61
		surprisal_0	0.12	0.17	0.71
		surprisal_1	0.40	0.17	2.38
		LCD_0	0.70	0.12	5.81
		LCD_1	0.33	0.12	2.71
	FFD	intercept	121.82	4.40	27.71
		frequency_0	-11.74	0.72	-16.36
		frequency_1	3.79	0.69	5.49
		length_0	25.07	0.59	42.59
		length_1	-9.27	0.58	-15.89
		surprisal_0	1.20	0.59	2.05
		surprisal_1	5.46	0.58	9.48
		LCD_0	-1.58	0.42	-3.78
		LCD_1	-1.98	0.43	-4.59
	GD	intercept	135.35	5.29	25.60
		frequency_0	-13.39	0.82	-16.39
		frequency_1	3.08	0.79	3.92
		length_0	31.65	0.67	47.22
		length_1	-13.10	0.66	-19.72
		surprisal_0	3.77	0.67	5.63
		surprisal_1	5.82	0.66	8.87
		LCD_0	-2.55	0.48	-5.36
		LCD_1	-2.23	0.49	-4.53
	GPT	intercept	148.67	5.40	27.53
		frequency_0	-14.43	0.96	-15.07
		frequency_1	2.66	0.92	2.89
		length_0	34.05	0.79	43.36
		length_1	-15.11	0.78	-19.42
		surprisal_0	4.29	0.78	5.47
		surprisal_1	7.60	0.77	9.90
		LCD_0	-4.23	0.56	-7.59
		LCD_1	-4.10	0.58	-7.12

Table 14: Detailed results for fitting a mixed linear effects model including surprisal from a small vanilla LM and leftmost connection distance as fixed effects, as well as our baseline predictors, with a spillover window of 1.

spill		effect	coef	Std. Error	t-value
leftmost connection distance and standard surprisal					
2	SPR	intercept	272.15	5.23	52.02
		frequency_0	0.19	0.21	0.88
		frequency_1	-2.11	0.22	-9.65
		frequency_2	-0.93	0.21	-4.42
		length_0	0.63	0.18	3.49
		length_1	1.05	0.18	5.82
		length_2	0.54	0.18	3.01
		surprisal_0	0.21	0.17	1.25
		surprisal_1	0.38	0.18	2.16
		surprisal_2	0.09	0.17	0.50
		LCD_0	0.62	0.13	4.90
		LCD_1	0.09	0.13	0.75
		LCD_2	0.27	0.13	2.12
	FFD	intercept	117.78	4.18	28.18
		frequency_0	-11.94	0.75	-15.92
		frequency_1	2.45	0.76	3.22
		frequency_2	-1.49	0.72	-2.05
		length_0	25.80	0.63	41.29
		length_1	-9.20	0.62	-14.84
		length_2	1.92	0.61	3.12
		surprisal_0	2.16	0.61	3.52
		surprisal_1	5.09	0.62	8.20
		surprisal_2	-1.28	0.61	-2.09
		LCD_0	-1.92	0.45	-4.28
		LCD_1	-1.84	0.45	-4.06
		LCD_2	1.22	0.46	2.65
	GD	intercept	128.44	5.00	25.67
		frequency_0	-13.51	0.83	-16.21
		frequency_1	2.09	0.85	2.48
		frequency_2	-2.27	0.80	-2.81
		length_0	31.99	0.69	46.09
		length_1	-11.16	0.69	-16.21
		length_2	2.03	0.68	2.98
		surprisal_0	3.02	0.68	4.44
		surprisal_1	5.40	0.69	7.83
		surprisal_2	-1.39	0.68	-2.04
		LCD_0	-1.79	0.50	-3.59
		LCD_1	-1.14	0.50	-2.27
		LCD_2	1.88	0.51	3.70
	GPT	intercept	139.36	5.01	27.81
		frequency_0	-13.58	0.96	-14.12
		frequency_1	3.04	0.98	3.11
		frequency_2	-2.35	0.93	-2.53
		length_0	34.25	0.80	42.72
		length_1	-13.36	0.80	-16.80
		length_2	0.62	0.79	0.78
		surprisal_0	3.51	0.79	4.46
		surprisal_1	7.13	0.80	8.95
		surprisal_2	-0.90	0.79	-1.14
		LCD_0	-3.19	0.58	-5.53
		LCD_1	-2.21	0.58	-3.79
		LCD_2	1.84	0.59	3.13

Table 15: Detailed results for fitting a mixed linear effects model including surprisal from a small vanilla LM and leftmost connection distance as fixed effects, as well as our baseline predictors, with a spillover window of 2.

	effect	coef	Std. Error	t-value
supervised surprisal				
SPR	intercept	272.52	5.26	51.80
	frequency_0	0.88	0.29	3.02
	frequency_1	-1.89	0.29	-6.47
	frequency_2	-1.01	0.27	-3.76
	length_0	1.05	0.24	4.35
	length_1	0.74	0.24	3.09
	length_2	0.50	0.23	2.19
	surprisal_0	0.38	0.21	1.83
	surprisal_1	0.96	0.22	4.33
	surprisal_2	0.10	0.20	0.47
FFD	intercept	117.87	4.22	27.92
	frequency_0	-12.62	1.01	-12.47
	frequency_1	4.78	1.00	4.79
	frequency_2	0.97	0.91	1.07
	length_0	25.56	0.85	30.11
	length_1	-8.41	0.83	-10.19
	length_2	3.40	0.81	4.18
	surprisal_0	2.67	0.76	3.53
	surprisal_1	4.70	0.80	5.87
	surprisal_2	-0.47	0.69	-0.68
GD	intercept	128.69	5.10	25.22
	frequency_0	-14.24	1.13	-12.64
	frequency_1	5.61	1.11	5.05
	frequency_2	1.06	1.01	1.04
	length_0	31.91	0.95	33.76
	length_1	-9.96	0.92	-10.84
	length_2	4.02	0.90	4.44
	surprisal_0	3.17	0.84	3.76
	surprisal_1	4.92	0.89	5.52
	surprisal_2	-0.61	0.77	-0.80
GPT	intercept	138.75	5.04	27.54
	frequency_0	-14.85	1.29	-11.53
	frequency_1	6.25	1.27	4.92
	frequency_2	1.43	1.16	1.24
	length_0	33.30	1.081	30.81
	length_1	-12.56	1.05	-11.95
	length_2	3.07	1.03	2.96
	surprisal_0	3.78	0.96	3.92
	surprisal_1	6.43	1.02	6.32
	surprisal_2	0.20	0.88	0.23

Table 16: Detailed results for fitting a mixed linear effects model including surprisal from our supervised model as a fixed effect, as well as our baseline predictors. Shifted predictors for spillover window sizes 1 and 2 are also included.

	effect	coef	Std. Error	t-value
predicted leftmost connection distance				
SPR	intercept	272.49	5.26	51.79
	frequency_0	0.71	0.27	2.61
	frequency_1	-2.55	0.26	-9.85
	frequency_2	-1.12	0.24	-4.59
	length_0	1.13	0.24	4.69
	length_1	0.87	0.24	3.65
	length_2	0.52	0.23	2.28
	PLCD_0	0.40	0.15	2.63
	PLCD_1	0.04	0.15	0.25
	PLCD_2	0.28	0.16	1.74
FFD	intercept	118.26	4.22	28.01
	frequency_0	-14.59	0.92	-15.82
	frequency_1	2.07	0.88	2.35
	frequency_2	1.31	0.84	1.56
	length_0	25.71	0.85	30.41
	length_1	-7.46	0.82	-9.16
	length_2	3.72	0.80	4.63
	PLCD_0	-0.41	0.55	-0.75
	PLCD_1	-2.15	0.54	-4.01
	PLCD_2	-0.00	0.62	-0.01
GD	intercept	129.00	5.10	25.28
	frequency_0	-16.40	1.03	-15.97
	frequency_1	2.75	0.98	2.80
	frequency_2	1.44	0.94	1.53
	length_0	32.14	0.94	34.15
	length_1	-8.92	0.91	-9.84
	length_2	4.31	0.89	4.82
	PLCD_0	0.07	0.61	0.11
	PLCD_1	-2.04	0.60	-3.41
	PLCD_2	0.14	0.70	0.20
GPT	intercept	139.48	5.04	27.68
	frequency_0	-17.75	1.17	-15.12
	frequency_1	2.59	1.12	2.31
	frequency_2	1.56	1.07	1.46
	length_0	33.47	1.08	31.10
	length_1	-11.28	1.04	-10.88
	length_2	3.70	1.02	3.62
	PLCD_0	-1.26	0.69	-1.82
	PLCD_1	-3.29	0.68	-4.81
	PLCD_2	-0.38	0.79	-0.48

Table 17: Detailed results for fitting a mixed linear effects model including leftmost connection distance predicted by our supervised model as a fixed effect, as well as our baseline predictors. Shifted predictors for spillover window sizes 1 and 2 are also included.

spill		effect	coef	Std. Error	t-value
predicted leftmost connection distance and supervised surprisal					
0	SPR	intercept	272.52	5.29	51.56
		frequency_0	1.29	0.21	6.08
		length_0	0.67	0.18	3.66
		surprisal_0	0.62	0.16	3.81
		PLCD_0	0.45	0.12	3.75
	FFD	intercept	122.11	4.78	25.56
		frequency_0	-14.65	0.72	-20.39
		length_0	23.63	0.62	38.25
		surprisal_0	1.44	0.56	2.56
		PLCD_0	-1.93	0.42	-4.61
	GD	intercept	136.99	5.89	23.26
		frequency_0	-17.51	0.83	-20.97
		length_0	31.51	0.72	43.89
		surprisal_0	3.87	0.65	5.92
		PLCD_0	-3.78	0.49	-7.80
	GPT	intercept	149.92	6.03	24.86
		frequency_0	-18.61	0.97	-19.23
		length_0	34.27	0.83	41.19
		surprisal_0	4.13	0.76	5.45
		PLCD_0	-4.75	0.56	-8.45

Table 18: Detailed results for fitting a mixed linear effects model including surprisal from our supervised model and predicted leftmost connection distance as fixed effects, as well as our baseline predictors, without any spillover.

spill		effect	coef	Std. Error	t-value
predicted leftmost connection distance and supervised surprisal					
1	SPR	intercept	272.15	5.28	51.50
		frequency_0	0.83	0.26	3.22
		frequency_1	-1.65	0.23	-7.08
		length_0	0.88	0.21	4.19
		length_1	1.11	0.20	5.56
		surprisal_0	0.34	0.19	1.74
		surprisal_1	0.44	0.18	2.48
		PLCD_0	0.57	0.13	4.47
		PLCD_1	0.36	0.14	2.56
	FFD	intercept	121.99	4.50	27.11
		frequency_0	-14.18	0.87	-16.33
		frequency_1	4.65	0.77	6.03
		length_0	24.50	0.70	34.90
		length_1	-7.20	0.68	-10.66
		surprisal_0	1.66	0.68	2.43
		surprisal_1	4.63	0.59	7.81
		PLCD_0	-1.18	0.45	-2.66
		PLCD_1	-1.63	0.50	-3.25
	GD	intercept	134.31	5.35	25.09
		frequency_0	-15.86	0.98	-16.19
		frequency_1	3.93	0.87	4.51
		length_0	31.05	0.79	39.20
		length_1	-9.82	0.76	-12.87
		surprisal_0	3.16	0.77	4.11
		surprisal_1	4.83	0.67	7.21
		PLCD_0	-1.19	0.50	-2.37
		PLCD_1	-1.82	0.57	-3.22
	GPT	intercept	147.92	5.45	27.15
		frequency_0	-16.95	1.15	-14.68
		frequency_1	3.27	1.03	3.19
		length_0	33.44	0.93	35.81
		length_1	-12.64	0.90	-14.06
		surprisal_0	4.26	0.91	4.71
		surprisal_1	6.00	0.79	7.61
		PLCD_0	-2.84	0.59	-4.79
		PLCD_1	-4.21	0.67	-6.32

Table 19: Detailed results for fitting a mixed linear effects model including surprisal from our supervised model and predicted leftmost connection distance as fixed effects, as well as our baseline predictors, with a spillover window of 1.

spill		effect	coef	Std. Error	t-value
predicted leftmost connection distance and supervised surprisal					
2	SPR	intercept	272.44	5.26	51.78
		frequency_0	1.02	0.30	3.44
		frequency_1	-1.93	0.29	-6.59
		frequency_2	-0.99	0.27	-3.69
		length_0	1.06	0.24	4.40
		length_1	0.70	0.24	2.94
		length_2	0.45	0.23	1.97
		surprisal_0	0.41	0.21	1.97
		surprisal_1	1.01	0.22	4.55
		surprisal_2	0.18	0.20	0.90
		LCD_0	0.46	0.15	2.96
		LCD_1	0.11	0.15	0.73
		LCD_2	0.33	0.16	2.01
	FFD	intercept	118.07	4.22	27.95
		frequency_0	-12.68	1.03	-12.35
		frequency_1	4.44	1.01	4.41
		frequency_2	0.96	0.91	1.06
		length_0	25.51	0.85	30.01
		length_1	-8.32	0.83	-10.07
		length_2	3.52	0.82	4.31
		surprisal_0	2.57	0.76	3.39
		surprisal_1	4.37	0.81	5.40
		surprisal_2	-0.63	0.70	-0.89
		PLCD_0	-0.24	0.55	-0.45
		PLCD_1	-1.70	0.54	-3.13
		PLCD_2	0.21	0.63	0.34
	GD	intercept	128.78	5.10	25.23
		frequency_0	-14.14	1.14	-12.36
		frequency_1	5.25	1.12	4.69
		frequency_2	1.06	1.01	1.05
		length_0	31.88	0.95	33.70
		length_1	-9.88	0.92	-10.75
		length_2	4.09	0.91	4.51
		surprisal_0	3.12	0.84	3.70
		surprisal_1	4.66	0.90	5.18
		surprisal_2	-0.65	0.78	-0.83
		PLCD_0	0.26	0.61	0.43
		PLCD_1	-1.54	0.60	-2.55
		PLCD_2	0.38	0.70	0.54
	GPT	intercept	139.20	5.04	27.61
		frequency_0	-15.15	1.31	-11.59
		frequency_1	5.81	1.28	4.53
		frequency_2	1.41	1.16	1.22
		length_0	33.17	1.08	30.67
		length_1	-12.39	1.05	-11.78
		length_2	3.30	1.04	3.18
		surprisal_0	3.57	0.97	3.69
		surprisal_1	5.86	1.03	5.70
		surprisal_2	-0.19	8.96	-0.21
		PLCD_0	-0.98	0.70	-1.41
		PLCD_1	-2.64	0.69	-3.82
		PLCD_2	0.00	0.80	0.000

Table 20: Detailed results for fitting a mixed linear effects model including surprisal from our supervised model and predicted leftmost connection distance as fixed effects, as well as our baseline predictors, with a spillover window of 2.

	effect	coef	Std. Error	t-value
GPT2 surprisal				
SPR	intercept	272.15	5.23	52.01
	frequency_0	0.15	0.20	0.75
	frequency_1	-1.91	0.20	-9.42
	frequency_2	-0.45	0.19	-2.37
	length_0	0.67	0.18	3.72
	length_1	1.00	0.18	5.60
	length_2	0.41	0.18	2.30
	surprisal_0	0.30	0.15	2.05
	surprisal_1	0.82	0.15	5.36
	surprisal_2	1.13	0.15	7.77
FFD	intercept	117.78	4.18	28.18
	frequency_0	-11.94	0.68	-17.47
	frequency_1	3.62	0.71	5.08
	frequency_2	-0.13	0.65	-0.20
	length_0	26.03	0.62	41.68
	length_1	-9.43	0.62	-15.28
	length_2	1.20	0.61	1.95
	surprisal_0	1.34	0.54	2.49
	surprisal_1	6.54	0.56	11.61
	surprisal_2	1.69	0.51	3.29
GD	intercept	128.44	5.01	25.66
	frequency_0	-13.58	0.76	-17.91
	frequency_1	3.33	0.79	4.22
	frequency_2	-0.88	0.72	-1.21
	length_0	32.05	0.69	46.22
	length_1	-11.39	0.69	-16.61
	length_2	1.41	0.68	2.07
	LCD_0	2.13	0.60	3.55
	LCD_1	6.86	0.63	10.96
	LCD_2	1.58	0.57	2.77
GPT	intercept	139.36	5.013	27.80
	frequency_0	-12.92	0.88	-14.75
	frequency_1	4.62	0.91	5.06
	frequency_2	-1.13	0.83	-1.36
	length_0	34.33	0.80	42.87
	length_1	-13.91	0.79	-17.57
	length_2	-0.28	0.79	-0.36
	LCD_0	3.41	0.69	4.93
	LCD_1	9.30	0.72	12.87
	LCD_2	2.24	0.66	3.40

Table 21: Detailed results for fitting a mixed linear effects model including surprisal predicted by GPT2 as a fixed effect, as well as our baseline predictors. Shifted predictors for spillover window sizes 1 and 2 are also included.

Syntax-Guided Parameter Efficient Fine-Tuning of Large Language Models

Prasanth Yadla
Independent Researcher
USA
pyadla2@alumni.ncsu.edu

Abstract

Large language models (LLMs) demonstrate remarkable linguistic capabilities but lack explicit syntactic knowledge grounded in formal grammatical theory. This paper introduces a syntax-guided parameter-efficient fine-tuning approach[1] that integrates formal syntactic constraints into transformer-based models using Low-Rank Adaptation (LoRA). We develop a hybrid training objective incorporating violations of syntactic well-formedness derived from dependency parsing and context-free grammar constraints. Our method is evaluated on established English syntactic benchmarks including BLiMP, CoLA, and SyntaxGym targeting specific grammatical phenomena. Results show modest but consistent improvements in syntactic competence: 1.6 percentage point average improvement on BLiMP overall, with gains of 1.7 percentage points on agreement phenomena and 1.6 percentage points on filler-gap dependencies, alongside 0.006 improvement in CoLA MCC scores, while maintaining stable performance on general natural language processing (NLP) tasks. The parameter-efficient approach reduces training time by 76% compared to full fine-tuning while achieving these incremental syntactic gains. This work demonstrates a practical pathway for incorporating linguistic theory into modern natural language processing (NLP) systems, though the improvements suggest that explicit syntactic supervision provides limited additional benefits over implicit learning from large-scale text.

1 Introduction

The extraordinary success of large language models (LLMs) in natural language processing has largely been achieved through statistical learning from massive text corpora, with minimal explicit incorporation of linguistic theory (Brown et al., 2020; Touvron et al., 2023). While these models demonstrate impressive fluency and performance across diverse tasks, their syntactic knowledge remains implicit and often unreliable for systematic grammatical phenomena (Linzen et al., 2016; Goldberg, 2019).

Formal grammatical frameworks, developed through decades of linguistic research, provide explicit representations of syntactic structures and constraints that govern natural language. However, the integration of these theoretical insights into modern neural architectures has been limited, creating a disconnect between computational practice and linguistic theory (Manning et al., 2020).

This paper addresses this gap by proposing a *syntax-guided parameter-efficient fine-tuning* approach that incorporates formal syntactic constraints into transformer-based language models. Our method leverages Low-Rank Adaptation (LoRA) (Hu et al., 2022) to efficiently integrate syntactic supervision while preserving the general capabilities of pre-trained models.

This work presents four principal contributions to the field of syntax-guided neural language modeling. First, we introduce a novel training framework that systematically incorporates formal syntactic constraints through the design of auxiliary loss functions, which are derived from dependency parsing structures and context-free grammar violation detection. Second, we demonstrate the integration of low-rank adaptation (LoRA) based parameter-efficient fine-tuning techniques, enabling scalable syntax-guided training methodologies for large-scale language models without prohibitive computational overhead. Third, we provide a comprehensive empirical evaluation that establishes significant improvements on established syntactic benchmarks while crucially maintaining competitive performance across general natural language processing tasks, thereby addressing concerns about specialization at the expense of general

[1] https://github.com/TransformerTitan/SyntaxGuidedPEFT

Proceedings of the Second Workshop on the Bridges and Gaps between Formal and Computational Linguistics (BriGap-2), pages 72–78

capability. Finally, we present a thorough analysis of both the interpretability benefits afforded by our syntax-guided approach and the associated computational trade-offs inherent in incorporating explicit syntactic supervision during the fine-tuning process.

2 Related Work

2.1 Syntactic Evaluation of Language Models

Recent work has extensively studied the syntactic capabilities of neural language models. Linzen et al. (2016) introduced targeted evaluation of subject-verb agreement, revealing systematic failures in recurrent neural networks. Warstadt et al. (2020) developed the BLiMP benchmark for comprehensive syntactic evaluation, showing that while transformers perform better than RNNs, significant gaps remain in syntactic competence.

Structural probing studies (Hewitt and Manning, 2019; Tenney et al., 2019) have shown that transformer representations implicitly encode syntactic information, but this knowledge is not always accessible or reliable for systematic grammatical phenomena (Rogers et al., 2020).

2.2 Neural-Symbolic Integration

Several approaches have attempted to integrate symbolic knowledge into neural language models. Kuncoro et al. (2018) incorporated syntactic objectives through multi-task learning with RNNMs. Strubell et al. (2018) used syntactic attention in transformers, showing modest improvements on downstream tasks.

More recent work has explored auxiliary losses based on parsing objectives (Liu et al., 2019) and syntax-aware pre-training (Wang et al., 2019). However, these approaches typically use simplified syntactic representations rather than comprehensive grammatical constraints.

2.3 Parameter-Efficient Fine-Tuning

Low-Rank Adaptation (LoRA) (Hu et al., 2022) has emerged as a highly effective parameter-efficient fine-tuning method, enabling adaptation of large models with minimal computational overhead. Dettmers et al. (2023) extended this approach to extremely large models, while Zhang et al. (2023) proposed adaptive rank allocation for improved efficiency.

Our work is the first to systematically combine LoRA with formal syntactic constraints, demonstrating that parameter-efficient methods can effectively incorporate linguistic knowledge.

3 Methodology

3.1 Formal Syntactic Constraints

We define formal syntactic constraints based on two primary sources of grammatical violations. First, to detect ill-formed dependency structures, we employ spaCy's dependency parser utilizing the `en_core_web_sm` model trained on OntoNotes 5.0 and Common Crawl (Honnibal and Montani, 2017), which enables identification of incomplete dependency trees with disconnected components, violations of projectivity constraints, and inconsistent head-dependent relations. Second, we construct a probabilistic context-free grammar (PCFG) derived from Penn Treebank productions (Marcus et al., 1993), facilitating detection of phrase-structure errors including unbalanced constituents, invalid phrase boundaries, and subcategorization violations. For each training sentence, we compute violation scores $v_{\text{dep}}(s)$ and $v_{\text{cfg}}(s)$ that quantify the severity of dependency-based and CFG-based violations, respectively.

3.2 Syntax-Guided Loss Function

To incorporate syntactic supervision into training, we extend the standard language modeling objective with penalties derived from the above constraints. The total loss is given by

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{LM}} + \alpha \mathcal{L}_{\text{syntax}}, \tag{1}$$

where $\mathcal{L}_{\text{LM}}$ is the conventional cross-entropy loss and α modulates the influence of syntactic penalties. The syntax-aware component is decomposed as

$$\begin{aligned} \mathcal{L}_{\text{syntax}} &= \mathcal{L}_{\text{dep}} + \mathcal{L}_{\text{cfg}}, \\ \mathcal{L}_{\text{dep}} &= \mathbb{E}_{s \sim D}\big[v_{\text{dep}}(s) \cdot \log P(s)\big], \\ \mathcal{L}_{\text{cfg}} &= \mathbb{E}_{s \sim D}\big[v_{\text{cfg}}(s) \cdot \log P(s)\big]. \end{aligned} \tag{2}$$

where D denotes the training distribution and $v_{\text{dep}}(s)$, $v_{\text{cfg}}(s)$ are violation functions that quantify dependency and context-free grammar violations, respectively. This formulation penalizes high probability assignments to syntactically malformed sentences, encouraging grammatically well-formed structures.

3.3 LoRA Integration

To achieve parameter-efficient fine-tuning, we integrate low-rank adaptation (LoRA) into the syntax-guided training framework. For each weight matrix

$W_0 \in \mathbb{R}^{d \times k}$ in the transformer, LoRA introduces a low-rank decomposition with trainable matrices $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times k}$, where $r \ll \min(d, k)$. The adapted weight matrix is expressed as

$$W = W_0 + \Delta W = W_0 + BA. \tag{3}$$

During fine-tuning, only the LoRA parameters $\{A, B\}$ are updated, while the original pre-trained weights W_0 remain frozen, significantly reducing the number of trainable parameters while preserving model expressivity. LoRA modifications are applied to the query, key, value, and output projection matrices in the attention layers, as well as to the up and down-projection matrices within the feed-forward networks.

4 Experimental Setup

4.1 Models and Baselines

We experiment with Llama 2-7B (7 billion parameters) and Mistral-7B (7.3 billion parameters) as base models, representing state-of-the-art open source architectures with strong performance across diverse tasks. Our comparison includes several baseline approaches to establish the effectiveness of syntax-guided training. The vanilla baseline uses pre-trained models without any fine-tuning to establish lower bounds on performance. We also compare against LoRA baseline fine-tuning that uses only language modeling loss without syntactic supervision.

4.2 Training Procedure

Our training procedure consists of two distinct phases designed to systematically incorporate syntactic knowledge into language models. The first phase involves syntactic annotation, where we process the training corpus through syntactic parsers to compute violation scores. Specifically, we utilize subsets of BookCorpus (Zhu et al., 2015) comprising 11,038 books (approximately 74M sentences) and OpenWebText (Gokaslan and Cohen, 2019) containing 8.01M web documents (approximately 40GB of text data), covering diverse domains including fiction, news articles, reference materials, and web content. This preprocessing step creates an augmented dataset enriched with syntactic constraint information that guides subsequent training.

The second phase implements LoRA fine-tuning (Hu et al., 2022), where we fine-tune pre-trained models including Llama 2-7B (7 billion parameters) (Touvron et al., 2023) and Mistral-7B (7.3 billion parameters) (Jiang et al., 2023) on the syntactically annotated BookCorpus and OpenWebText subsets using our syntax-guided loss function. LoRA rank r is set to 16 for attention layers and 32 for feed-forward layers based on preliminary experiments that balanced computational efficiency with representational capacity. Training is conducted for 3 epochs with gradient accumulation steps of 8 to effectively utilize the available computational resources.

Hyperparameters are systematically tuned on held-out validation sets comprising 10% of the training data to ensure optimal performance. We explore loss weighting values $\alpha \in \{0.1, 0.5, 1.0, 2.0\}$ to balance syntactic supervision with language modeling objectives. Learning rates are tested across $\{1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$ to determine optimal optimization dynamics, while batch sizes are evaluated over $\{16, 32, 64\}$ to maximize training stability and convergence speed.

4.3 Violation score computation

The dependency violation score $v_{\text{dep}}(s)$ is computed by applying the spaCy dependency parser to sentences from our training corpora (BookCorpus and OpenWebText subsets) and quantifying structural irregularities in the resulting parse trees. We interpret parser uncertainty and structural anomalies as indicators of potential grammatical issues, following the principle that well-formed sentences should yield clean, confident parses. Specifically, we assess *connectivity violations* by identifying cases where spaCy produces fragmented dependency structures due to parsing failures or ambiguity, computing $c_{\text{conn}}(s) = \frac{|\text{disconnected components}|}{|s|}$ when the parser cannot establish a fully connected tree. We detect *projectivity violations* by examining the confidence scores and alternative parse hypotheses from spaCy's beam search, where lower confidence in the primary parse or high-scoring non-projective alternatives indicate potential structural issues: $c_{\text{proj}}(s) = 1 - \text{confidence}_{\text{primary parse}}(s)$. Additionally, we evaluate *consistency violations* by flagging dependency relations that receive low probability under spaCy's statistical model, computed as $c_{\text{cons}}(s) = \frac{\sum_{(h,d,r) \in \text{parse}(s)} \mathbb{I}[P_{\text{spaCy}}(r|h,d) < \tau]}{|\text{dependencies}|}$, where τ is a threshold for acceptable relation confidence. The final dependency violation score combines these measures as $v_{\text{dep}}(s) = 0.4 \cdot c_{\text{conn}}(s) + 0.4 \cdot c_{\text{proj}}(s) + 0.2 \cdot c_{\text{cons}}(s)$.

The context-free grammar violation score $v_{\text{cfg}}(s)$

Task	No fine-tuning	LoRA	Syntax-Guided LoRA
BLiMP (Overall)	69.2	70.1	**70.8**
Agreement	72.4	73.2	**74.1**
Filler-Gap	64.1	65.0	**65.7**
Islands	61.3	62.1	**62.5**
Binding	75.2	76.0	**76.9**
CoLA (MCC)	0.448	0.453	**0.459**
SyntaxGym	66.7	67.2	**68.1**

Table 1: Results on syntactic evaluation benchmarks. Scores are accuracy (%) except CoLA which reports Matthews Correlation Coefficient.

is computed by parsing training corpus sentences with a PCFG extracted from Penn Treebank and using parse probability as a proxy for grammatical well-formedness. We extract production rules and their frequencies from the Penn Treebank to construct a probabilistic grammar, then attempt to parse each training sentence s with this grammar. The primary violation measure is *parse probability*, where sentences receiving low probability under the PCFG are considered potentially ungrammatical: $c_{\text{parse}}(s) = \max\left(0, \frac{\theta - \log P_{\text{PCFG}}(s)}{Z}\right)$, where $P_{\text{PCFG}}(s)$ is the probability of the best parse, $\theta = -10$ represents a grammaticality threshold empirically determined from well-formed sentences, and $Z = 20$ normalizes scores to $[0, 1]$. Sentences that cannot be parsed at all receive the maximum violation score of 1.0. We also compute *subcategorization violations* by checking whether the PCFG parse satisfies basic argument structure requirements, flagging cases where transitive verbs lack objects or other clear subcategorization violations: $c_{\text{subcat}}(s) = \frac{|\text{subcategorization violations in parse}(s)|}{|\text{verbs in } s|}$. The final CFG violation score is $v_{\text{cfg}}(s) = 0.7 \cdot c_{\text{parse}}(s) + 0.3 \cdot c_{\text{subcat}}(s)$. Both violation scores serve as continuous measures of grammatical deviance, with higher scores indicating sentences that our syntactic analyzers consider less well-formed, thereby providing supervision signal to discourage the language model from assigning high probability to potentially ungrammatical text.

4.4 Evaluation Benchmarks

Our evaluation focuses on both syntactic understanding and general language capabilities. For syntactic assessment, we employ BLiMP (Warstadt et al., 2020), which contains 67 sub-tasks testing various grammatical phenomena through minimal pairs that isolate specific syntactic knowledge. The CoLA benchmark (Warstadt et al., 2019) provides binary acceptability judgments on 10,657 sentences, testing broad grammatical competence. SyntaxGym (Gauthier et al., 2020) offers targeted evaluation using surprisal-based metrics that assess fine-grained syntactic processing capabilities.

For general language understanding evaluation, we utilize the GLUE benchmark tasks (Wang et al., 2018) to ensure that syntactic improvements do not compromise broader natural language processing capabilities across diverse tasks including sentiment analysis, textual entailment, and semantic similarity. We also assess text generation quality through perplexity measurements on WikiText-103 (Merity et al., 2017) and evaluate reading comprehension performance using SQuAD 2.0 (Rajpurkar et al., 2018) to capture the model's ability to process and understand complex textual information beyond syntactic parsing.

4.5 Evaluation Metrics

For syntactic tasks, we report accuracy on minimal pair judgments and Matthews Correlation Coefficient (MCC) for CoLA, providing robust measures of grammatical competence. For general tasks, we employ task-specific metrics including accuracy, F1 score, and perplexity as appropriate. We also measure training efficiency in terms of wall-clock time and GPU memory usage to demonstrate the practical viability of our approach.

5 Results

5.1 Syntactic Performance

Table 1 shows results on key syntactic benchmarks. Our syntax-guided LoRA approach achieves consistent improvements across all evaluated phenomena, with particularly notable gains in complex grammatical constructions.

The syntax-guided approach demonstrates modest but consistent improvements on agreement phenomena, achieving gains of 1.7 percentage points, and filler-gap dependencies with improvements of

Task	No fine-tuning	LoRA	Syntax-Guided LoRA
GLUE Average	83.2	83.6	83.4
SST-2	94.1	94.3	94.2
MRPC	89.2	89.7	89.1
QQP	91.8	92.1	92.0
MNLI	86.4	86.8	86.5
QNLI	91.3	91.7	91.4
RTE	69.1	70.2	69.8
WikiText-103 PPL	21.8	21.4	21.6
SQuAD 2.0 F1	82.3	82.9	82.7

Table 2: Performance on general NLP tasks. GLUE scores are accuracy (%), WikiText-103 is perplexity (lower is better), SQuAD 2.0 is F1 score.

1.6 percentage points. These results suggest that explicit syntactic constraints provide incremental benefits for challenging grammatical constructions, though the improvements are relatively small, indicating that such phenomena remain difficult for models to master even with targeted supervision.

5.2 General NLP Performance

Table 2 demonstrates that the syntax-guided approach maintains general language capabilities with minimal impact. While most GLUE tasks show small variations within typical noise margins, the overall GLUE average remains stable, indicating that the syntactic modifications do not significantly compromise broader language understanding. The slight variations across individual tasks suggest that syntactic constraints introduce minor trade-offs rather than uniform improvements, which is consistent with specialization effects observed in targeted fine-tuning approaches.

5.3 Computational Efficiency

Table 3 compares the computational requirements of different fine-tuning approaches, demonstrating that our method maintains the efficiency advantages of parameter-efficient training while incorporating valuable syntactic knowledge.

The syntax-guided approach adds minimal computational overhead compared to standard LoRA, requiring only approximately 16% additional training time while achieving substantial efficiency gains over full fine-tuning. The modest increase in memory usage reflects the additional syntactic constraint processing without fundamentally altering the parameter-efficient nature of the approach.

6 Analysis and Discussion

6.1 Qualitative Analysis

We analyze model outputs to understand the nature of syntactic improvements achieved through our approach. Examples demonstrate enhanced consistency in complex agreement patterns that frequently challenge standard language models. The baseline model produces: *"The collection of books that was donated by the students were placed on the shelf."* In contrast, our syntax-guided model correctly generates: *"The collection of books that was donated by the students was placed on the shelf."* This example illustrates how the syntax-guided model correctly maintains singular agreement with the head noun "collection" despite the presence of the plural intervening noun "students," a challenging construction that often leads to agreement errors.

6.2 Interpretability Benefits

The explicit incorporation of syntactic constraints enhances model interpretability in several meaningful ways. Syntactic violations can be traced to specific grammatical constraints that were violated during generation, providing clear diagnostic information about model failures. Attention patterns show improved alignment with syntactic structure, making it easier to understand how the model processes grammatical relationships. Additionally, model confidence correlates more strongly with grammatical acceptability, suggesting that syntactic training helps calibrate the model's uncertainty estimates.

6.3 Limitations

Our approach faces several limitations that constrain its applicability and effectiveness. The dependence on parser quality limits effectiveness when processing noisy or non-standard text, as parsing

Method	Trainable Params	Training Time	Memory (GB)
Full Fine-tuning	7.0B (100%)	156.3 hours	48.2
LoRA	41.9M (0.60%)	31.7 hours	18.4
Syntax-Guided LoRA	41.9M (0.60%)	36.8 hours	19.1

Table 3: Computational efficiency comparison for Llama 2-7B. Training time measured on 8×A100 GPUs for one epoch on our training corpus.

errors propagate through the training process. Computational overhead during training arises from the need for syntactic annotation and constraint processing, though this remains manageable within the parameter-efficient framework. The current focus on English syntax limits cross-lingual applicability, though the general framework could potentially be extended to other languages with appropriate syntactic resources.

7 Conclusion and Future Work

This paper demonstrates that formal syntactic constraints can be effectively integrated into large language models through parameter-efficient fine-tuning. Our syntax-guided LoRA approach achieves consistent improvements on syntactic benchmarks while maintaining general NLP performance and computational efficiency.

The key insights from this work demonstrate that explicit syntactic supervision provides complementary benefits to implicit learning from text, enabling models to achieve more robust grammatical competence. Parameter-efficient methods enable scalable integration of linguistic constraints without the computational burden of full model retraining. Furthermore, formal grammatical knowledge enhances both performance and interpretability, making models more reliable and diagnostic.

Future work should explore extension to multilingual models and diverse syntactic frameworks, particularly investigating how different grammatical traditions and linguistic theories can be incorporated into modern architectures. Integration with other parameter-efficient methods such as AdaLoRA and prefix tuning could potentially yield additional benefits. Application to semantic and pragmatic constraints beyond syntax represents a natural extension of this work. Finally, investigation of emergent syntactic capabilities in very large models like GPT-4 and PaLM could reveal whether explicit syntactic guidance remains beneficial at scale.

This work provides a concrete pathway for reintegrating linguistic theory into modern NLP systems, suggesting that the future of language modeling may benefit from renewed collaboration between formal linguistics and computational practice.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sasuke, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. Syntaxgym: An online platform for targeted evaluation of language models. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–75.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext: An open source recreation of the gpt-2 training dataset.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4129–4138.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. In *Proceedings of the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 411–420.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego

de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A Smith. 2018. Lstms can learn syntax-sensitive dependencies well, but modeling structure makes them better. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1426–1436.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew E Peters, and Noah A Smith. 2019. Linguistic knowledge and transferability of contextual representations. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1073–1094.

Christopher D Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 2, pages 784–789.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Emma Strubell, Patrick Verga, Daniel Belanger, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R 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.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Jiangnan Xia, Liwei Peng, and Luo Si. 2019. Structbert: Incorporating language structures into pre-training for deep language understanding. *arXiv preprint arXiv:1908.04577*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Qingru Zhang, Minshuo Zuo, Denghui Zhou, Quanquan Mei, and Hao Chen. 2023. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*.

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.

On the relative impact of categorical and semantic information on the induction of self-embedding structures

Antoine Venant
OLST, Université de Montréal
antoine.venant@umontreal.ca

Yutaka Suzuki
OLST, Université de Montréal
yutaka.suzuki@umontreal.ca

Abstract

We investigate the impact of center embedding and selectional restrictions on neural latent tree models' tendency to induce self-embedding structures. To this aim we compare their behavior in different controlled artificial environments involving noun phrases modified by relative clauses, with different quantities of available training data. Our results provide evidence that the existence of multiple center self-embedding is a stronger incentive than selectional restrictions alone, but that the combination of both is the best incentive overall. We also show that different architectures benefit very differently from these incentives.

1 Introduction

Grammar induction is the task of inducing hierarchical syntax trees from *indirect* observations, most often provided by the string yields of those trees (raw sentences)[1]. The most common type of approach parametrizes a joint distribution over (observed) strings and (latent) trees, and fit it to the data by optimizing a language modeling objective, *i.e.* minimizing the cross entropy between the model's marginal distribution over strings and the observed distribution. We will refer to approaches of this kind as *latent tree models*. Latent tree models are interesting because they can provide distributional evidence for (or counter arguments to) the structures stipulated by linguistic theories, help investigate inductive biases and language model pretraining, or build bridges between neural models and symbolic ones.

Grammar induction, however, is a rather difficult task. For the longest time, models were mostly trained and tested on very short sentences of about 10 words, struggled to beat baselines such as right- or left-linear grammars (Carroll and Charniak, 1992), heavily relied on heuristics for clustering (Clark, 2001), initialization (Klein and Manning, 2001, 2002, 2004) and/or assumed part-of-speech tagged inputs (Bisk and Hockenmaier, 2013). Whereas several potential culprits like an ill-shaped objective function (Klein and Manning, 2001), or data quantity (Pate and Johnson, 2016) have been named, a comprehensive explanation of the underlying difficulties is still lacking.

More recently, neural grammar induction models surfaced (Shen et al., 2018; Htut et al., 2018; Shen et al., 2019; Kim et al., 2019b,a; Yang et al., 2021; Zhu et al., 2020), which substitute the discrete features of their predecessors with continuous representations of input words and models' states, and parametrize the joint probability over tree structures and strings using a neural network. While these models considerably improved the state of the art for phrase structure induction from words alone, there remains a large gap between their performances and those of supervised parsers. An important question is **why a language modeling objective would align well with some (let alone all) of the intended structural patterns**. Moreover, the answer to this question might well be negative, because better unsupervised parsers are often worse language models and *vice versa* (Kim et al., 2019a).

In hope to improve our understanding of the matter, this paper investigates the combinations of training signal and neural models able to induce self-embedding structures, and the generalization capabilities of the learned models to larger phrases. We focus on the case of noun phrases, whose linguistic analysis is tied to phenomena such as (long-distance) subject-verb agreement, commonly used, across different languages, in benchmarking language models' syntactic awareness (Linzen et al., 2016; Marvin and Linzen, 2018; Li et al., 2023). We are specifically interested in the relative impact

[1] At least in one of its common usages in machine learning. Other usages include the task of inferring a specific form of formal grammar, or a recognizer from example sentences of a language.

Proceedings of the Second Workshop on the Bridges and Gaps between Formal and Computational Linguistics (BriGap-2), pages 79–96

of **categorical** incentives based on the sequences of coarse-grained part of speech categories, and lexical incentives based on more fine-grained **semantic** distinctions between words within a given category.

Since the complexity of natural language makes it hard to study specific aspects of the training signal and output structures in isolation, we experiment with artificial data generated with probabilistic grammars. This allows us to control for the presence of different incentives in the training signal, as well as the quantity of available training data. It also guarantees that at least one optimal model exists which leverages the intended structures. Unlike other works evaluating grammar induction systems on formal languages (Lari and Young, 1990; Lan et al., 2022), we focus on the *strong* learning of the intended structures, and use larger grammars with a sizeable lexicon to make learning syntactic categories and representing lexical features an integral (and non-trivial) part of the task. We do not consider alternative objectives (such as Minimum Description Length), because we precisely want to assess to what extent the (currently) more scalable language modeling objective aligns with theoretical patterns, if it does.

In §2 we discuss two incentives for inducing a self-embedding analysis of noun-phrases and relative clauses (henceforth, RC). §3 then presents how these incentives are implemented into four different artificial training signals. §4 details the experimental setup leveraging these data, and §5 discusses our findings and conclusions.

2 Why would a language model build noun phrases?

2.1 Distributional considerations

Linguists argue (across a variety of languages) that a noun, like *people*, can merge with a restrictive modifier, like *in a blue shirt* to form a noun phrase (NP), like *people in a blue shirt* (*e.g.* Baker, 1995; Tellier, 2003, for English and French, respectively). This would for instance happen twice when forming the English sentence *these* $[_{\text{NP}}$ *people in a* $[_{\text{NP}}$ *blue shirt with a collar*]] *are staff members*.

This analysis is often justified by a similarity in distribution between longer sequences (*people in a blue shirt*) and shorter ones (*people*). For instance, both are good candidates to fill the blanks in the following context: *these __ are staff members*.

It is thus compelling to assume that *people in a blue shirt with a collar* forms a constituent which inherits the morphosyntactic features (in particular, the number) and combinatorial properties of its head (*people*), because it explains why it combines, and agrees, with verbs as the bare noun *people* does. An important contribution of the hierarchical structure to that argument, is that it brings heads and dependents (the verb and subject in the above example) closer by grouping the intervening material inside a substructure who contributes little to the purpose of predicting agreement or the surounding context, and can therefore be pruned[2]. Formally, this process can (for instance) be equivalently articulated in a dependency framework, or in a headed constituency framework (Eisner and Satta, 1999; Nederhof and Satta, 2011).

2.2 An expected empirical difficulty

We have however no theoretical insurance that considerations like the above are sufficient for latent tree models to succeed. An important problem is raised by Klein and Manning (2002): the distribution of contexts in which a sequence occur is not necessarily a good indicator as to whether it is a constituent or not. Consider these two sequences:

A *the student who frequently questions the professor caused a problem*

B *the professor caused a problem*

A and B could plausibly occur in a lot of similar contexts, in which there are indeed constituents of the same type[3]. However, considered as a subsequence of A, B is not a constituent under any linguistic standard.

The problem is emphasized if one expects models to leverage a rather coarse notion of syntactic category (such as POS), because we can easily imagine such models to learn a **right-linear** grammar with a rule $S \rightarrow$ `det noun who verb` S. This grammar would generate sentences and parses such as $[_S$ *The student who questions* $[_S$ *the professor who questions* $[_S$ *the professor caused a problem*]]]. While we might find the induced string language somewhat reasonable, treating RCs as embedding sentence types is linguistically very unconventional.

[2] Or receive less attention, in a more relaxed, continuous vision of sentence processing.

[3] For instance in the context *Do you know whether ... ?*: *C*:*Do you know whether the student who frequently questions the professor caused a problem? D*:*Do you know whether the professor caused a problem?*.

2.3 Two possible incentives

Of course, an accurate language model needs to leverage more than categorical information. Thus, we might hope that semantic concerns, such as **selectional restrictions** (or, selectional *preferences* in a probabilistic view), can help break unwanted symmetry. For instance, the respective contexts of A': *the student who eats a cookie with sugar sprinkles passed the test* and B': *a cookie with sugar sprinkles passed the test* are probably more distinguished than those of A and B since, unlike B, B' is very unlikely to appear as a standalone sentence, as *cookie* violates the selectional restrictions of the verb *pass*. If a model is able to find a middle ground between relying on purely categorical and semantic knowledge, then we can hope that i) it makes a symmetric treatment of A and A' (based on categorical similarity) and ii) it makes the expected analysis of A' (based on semantic knowledge), hence of A (based on i). Moreover, selectional restrictions introduce a rich diversity of long-distance dependencies (such as between *student* and *pass* in A'), thus reinforcing the linguistic argument developed above.

Independently, it has been argued (Chomsky, 1956; Partee et al., 1990) that the set of 'grammatical' sentences of many languages are not representable by a right- or left-linear grammar because it involves **unbounded center-embedding** (or equivalently, unbounded well-nested projective dependencies). In English or French, this can for instance be argued through self-embedding of object RCs: *the student [that the professor [that your friend [(that . . .)] had]] dislikes] passed the test*. The set of sentences of this form is related[4] to the formal language $\{a^n b^n \mid n \in \mathbb{N}^*\}$, a canonical example of non-regular set. The possibility that this empirically affects latent tree models might however seem more remote, because it rests on an abstract notion of competence and an infinite set of sentences of arbitrary complexity, most of which are not attested (Karlson, 2007) or are rejected by speakers (Christiansen and MacDonald, 2009). Nevertheless, levels of center-embedding below four are attested (Karlson, 2007). Jin et al. (2018) introduced a grammar-based system that outperformed its contemporary competitors on English unsupervised parsing, and found that their system also achieved better performances on synthetic data with bounded center embeddings. This suggests that the ability to infer such bounded depth center-embedding, even on synthetic data with a very small lexicon, might be important for achieving good unsupervised parsing performances on natural language benchmarks. Moreover, state of the art neural language models' architectures have been theoretically and empirically demonstrated able to learn such bounded-depth occurrences (Yao et al., 2021).

Based on the above discussion, our objective will therefore be to assess the respective effect of **selectional restrictions** and **multiple center embedding** on different latent models.

3 Artificial Data

3.1 Target self-embedding

Our experiments are built around examples of self-embedding provided by French noun-phrases modified by subject, or object RC. These structures feature two types of self-embedding: final self-embedding (when the embedding constituent does not yield any material to the right of the embedded constituent), and center self-embedding (when the embedding constituent yields material to the left and right of the embedded constituent). In Figure 1, [$_{NP}$ *journaliste qui cherche le* [$_{NP}$ *succès*]] illustrates a case of final self-embedding, whereas [$_{NP}$ *article que le journaliste qui cherche le* [$_{NP}$ *succès*] *écrit*] illustrates a case of center self-embedding. In particular, subject and object RC respectively involve final and center self-embedding of NP. Additionally, RCs nested within object RC involve center self-embedding of CP and RCs nested within subject RCs involve final self-embedding of CP.

3.2 Four types of training signals

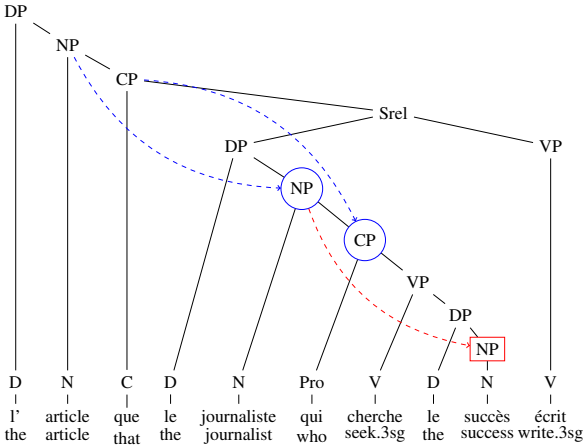
Our training data is artificially generated using PCFG (Probabilistic Context-Free Grammar). We first describe the (ideal) defining properties of the different training signals compared in our experiments, postponing the implementation's specifics to §3.3. All signals involve simple transitive or oblique sentences, with exactly one direct or exactly one oblique object. Every noun can be modified by exactly one RC which ensures that the data features examples of self-embedding.

This common basis is declined into four different

[4] Formally: $a^n b^n$ is an homomorphic image of the considered set of SVO sentences with arbitrary nesting of relative object in the subject position, and it follows from well-known closure properties that the former is regular (and thus represented by a right- or left-linear grammar) only if the latter (proven not regular) is.

-sr -mce	-sr +mce	+sr -mce	+sr +mce	sentence
1	1	1	1	le projet qui intéresse le journaliste qui écrit l'article présente un progrès 'the project which interesses the journalist who writes the article displays progress'
1	1	0	0	le projet qui parle au journaliste qui mange l'article regarde un progrès 'the project which talks to the journalist who eats the article watches progress'
0	1	0	1	le projet auquel le journaliste qui écrit l'article contribue présente un progrès 'the project to which the journalist who writes the article contributes displays progress'
0	1	0	0	le projet auquel le journaliste qui mange l'article parle regarde un progrès 'the project to which the journalist who eats the article talks watches progress'

Table 1: Type of sentences and compatibility with each configuration.



'the article that the journalist who seeks success writes'

Figure 1: French example with phrase structure tree and English gloss. Dashed edges indicate self-embedding, blue circles center embedded constituents and red squares final embedded constituents.

signals, depending on whether or not we simulate the two incentives discussed in the previous section. **-sr** means that the data does not simulate **selectional restrictions** in the probabilistic sense that all verbs, prepositions and nouns are conditionally independent of other words given their POS. This condition should thus assign equal probability to *article* (article) and *journaliste* (journalist) as subject of *écrire* (write). **+sr**, in contrast, means that the verb more likely selects semantically plausible subjects and objects. **+/-mce** means that the data exhibit / does not exhibit **multiple center-embedding**, according to whether noun phrases can be modified by either an (oblique) object RC or subject RC / can only be modified by a subject RC. Remark that the **-sr** and **+mce** configurations respectively *allow* selectional preference violations and object RCs, but do not *require* these to occur in every sentence. Thus, one may still observe some semantically sound sentences in the **-sr** configurations if appropriate verbal arguments are randomly selected (though, this will be unlikely if there are more lexical items violating a verb's selectional preferences than items respecting them), as well as sentences without multiple center-embeddding in the **+mce** configuration. Table 1 presents four example sentences and their compatibility with the four configurations: the first sentence neither violates selectional preferences nor features multiple center-embedding, hence could be observed under all four training configurations. The second sentence violates selectional preferences, and should thus only be observed in the two **-sr** configurations. The third sentence does not violate selectional preferences, but has multiple clausal and NP center-embedding due to the oblique object RC. It therefore can be observed only in the two **+mce** configurations. The fourth sentence both violates selectional preferences and has multiple center-embedding, and can therefore only be observed in the **-sr+mce** configuration. Note however, that table 1 presents a somewhat idealized picture: in practice, because the lexicon is acquired semi-automatically and the data generated probabilistically (see §3.3), selecting an inappropriate verbal argument will be unlikely rather than strictly impossible in the **+sr** configurations.

In line with the discussion in §2.2, we expected **-sr-mce** to provide the weakest signal for the induction of self-embedding structures and the condition **+sr+mce** to provide the strongest incentives.

3.3 Data generation

We used probabilistic context free grammars to generate the data for each configuration. More specifically, we used *lexicalized* PCFGs (Nederhof and Satta, 2011, henceforth LPCFG) to encode selectional preferences. Rather than the standard bilexical LPCFG, we used trilexical ones, with up to two anchors per nonterminal (this allows to keep the verb, preposition and object interdependent in oblique object constructions). We handcrafted *delexicalized* rules with anchor variables like the

following (used for the generation of a preposition x_1 and oblique object x_2, conditionally to a verb x_0):

$$\text{Vobl}'_{\langle x_0 \rangle} \mapsto \text{Vobl}_{\langle x_0 \rangle} \text{PP}_{\langle x_1, x_2 \rangle} [\text{o\_obj}(x_1, x_2 \mid x_0)] \quad (1)$$

In this example, the symbols Vobl', Vobl, PP are called delexicalized nonterminals and the symbols x_0, x_1 and x_2 are variables used as placeholders for terminal symbols. The expression $\text{o\_obj}(x_1, x_2 \mid x_0)$ is an *abstract weight*, formally representing a function associating a real-valued weight to concrete values for the variables x_0, x_1 and x_2 (in this case, the joint conditional probability of preposition and object, given verb).

To generate data, a delexicalized grammar needs to be combined with a set of terminal symbols, and **concrete** functions instantiating the abstract weights (henceforth, a *lexicon*). This allows to turn each delexicalized rule into a set of concrete rules. For instance, assuming that $\text{o\_obl}(\text{to}, \text{journalist} \mid \text{talk}) = 0.3$ and $\text{o\_obl}(\text{about}, \text{project} \mid \text{talk}) = 0.7$, the rule in (1) would yield the following *lexicalized* rules:

$$\text{Vobl}'_{\langle \text{talk} \rangle} \mapsto \text{Vobl}_{\langle \text{talk} \rangle} \text{PP}_{\langle \text{to,journalist} \rangle} [0.3] \quad (2)$$

$$\text{Vobl}'_{\langle \text{talk} \rangle} \mapsto \text{Vobl}_{\langle \text{talk} \rangle} \text{PP}_{\langle \text{about,project} \rangle} [0.7] \quad (3)$$

The resulting LPCFG can then be used to generate sentences with both a constituency structure and a dependency structure[5].

The remaining difficulty is to craft a lexicon sufficiently large for models to be 'forced' into some kind of categorization, while reasonably simulating the `+sr` configuration. To achieve this, we leveraged CamemBERT (Martin et al., 2020), a French masked language model. We manually fixed the sets of functional categories (prepositions $\mathcal{P}$ and determiners $\mathcal{D}$), as well as two sets of 34 transitive verbs and 20 intransitive oblique verbs. We then bootstrapped a lexicon of nouns and probability distributions using CamemBERT. To this effect, we made a set of requests to the masked language model. Let us examplify this with *parler* (talk). To obtain both oblique object ($\text{o\_obl}(x_1, x_2 \mid \text{parler})$ above) and subject probabilities, we used a set of masked requests of the form

(4) d_1 <mask>$_s$ parle x_1 d_2 <mask>$_o$

where $d_{1/2}$ range over determiners and x_1 over prepositions. For instance, $d_1 =$ *un*, $x_1 =$ *à* and $d_2 =$ *la* corresponds to the masked request:

(5) un <mask>$_s$ parle à la <mask>$_o$
a.m <mask>$_s$ talk.prs.3sg to the.f <mask>$_o$

For each request, CamemBERT outputs two conditional distributions $P_{s/o}(x_2 \mid d_1, x_1, d_2, \text{parler})$, one for each of the two masked positions (subject and object). From there, we simply assumed uniform prior and marginalized over any extra variable (like the determiners). For instance, we obtained $\text{o\_obl}(x_1, x_2 \mid \text{parler})$ by computing $\frac{1}{|\mathcal{D}|^2|\mathcal{P}|} \sum_{d_1,d_2} P_o(x_2 \mid d_1, x_1, d_2, \text{parler})$. We proceeded similarly for the other verb-noun distributions, then performed some additional filtering, keeping only the top 100 nouns for each conditional distribution, and only nouns that are both subject and object of some verbs (to ensure that they support both kind of modification by RC), and finally re-normalizing. After normalization, the resulting sentences are inflected using the French and English surface realizer `PyRealB` (Molins and Lapalme, 2015; Lapalme, 2020).

To simulate the four configurations, we combined the same delexicalized grammar with different concrete weights. The above procedure yielded a lexicon of 951 nouns, as well as the verb selectional distributions for the `+sr` configuration. For the `-sr` configuration, we replaced these distributions with uniform distributions on the relevant domains. In both cases, we used a uniform distribution for the choice of the main verb, and relied on Bayes' theorem to generate the verb in object and subject RC depending on the modified noun. In all configurations, we set a fixed probability (0.3) for modifying each noun. The nesting of RCs on a given noun thus follows a geometric law, and the number of generated sentences decays exponentially with the depth of nested RCs. Each training dataset therefore contains very rare instances of deeply nested embeddings. The grammar used for `+mce` condition has equal chances of attaching an object and subject RC, while the one for `-mce` only attaches subject RC. The probability of attachment and the expected degree of nesting are controlled and remain the same across configurations.

Note that, in the `-sr` configurations, the lexical anchors of the LPCFG can be safely deleted without changing the language (the anchors' only purpose is to implement selection restrictions). This operation leaves grammars in Chomsky Normal

[5] For dependency structures, a few adaptations are needed from the bilexical to trilexical case. Since the paper focuses on constituency, we do not expand on these technical matters.

Form with less than 30 nonterminal symbols. In contrast, in the +sr configurations, the generating grammars have over 17000 nonterminals, most of which are probably[6] necessary. In addition, these conditions involve long-distance dependencies between (at least) the subject of the main clause and the main verb. Note also, that both **-mce** configurations make it theoretically possible to perfectly fit the ground-truth distribution with a right-linear PCFG or a left-linear PCFG whereas the **+mce** conditions theoretically require branching structures for a perfect fit.

We might question whether our implementation of the +sr condition matches its definition, given the semi-automatic acquisition of the lexicon. However, looking at the verb-argument distributions, we found that the top subject and objects are generally semantically sound. For instance, the top 5 subjects of *investir* (*invest*) are *groupe* (*group*), *banque* (*bank*), compagnie (*company*), region (*region*) and *ville* (*city*) (covering about 75% of the probability mass), while the top 5 subject of *eat* are *femme* (*woman*), *chien* (*dog*), *homme* (*man*), *fille* (*girl*) and *chat* (*cat*) (covering about 40% of the probability mass). We also checked that the overall distribution of nouns follows Zipf's law, and estimated the mutual information between subject and main verb from the generated data to be approximately 2 bits (against 0 in -sr configurations).

4 Experiments

4.1 Training and test data

Using the procedure described in § 3.3, we generated over one million sentences for each of the four configurations and removed duplicates. The remaining data were split into training and development sets. From each training set, we constructed four subsets of approximately 3k, 12k, 100k, and 400k sentences by recursive halving, ensuring that all smaller subsets are prefixes of the larger ones (e.g., the first 3k sentences appear in all four datasets). This resulted in 16 training sets and four development sets of 3k sentences each.

Since the models trained under -mce never observe object RC, it would be unfair to compare their ability to parse sentences with object RC to model which have. We thus mainly compare models on their performance on data from the -mce configuration, *i.e.* their ability to analyze noun phrases modified by subject RC. We use the +sr-mce configuration for evaluation, because none of the four configurations are biased against any of its sentences: sentences from the +sr-mce are as likely as any other sentence with the same sequence of POS to occur in the -sr training data, whereas the converse is not true. We therefore generated an (out-of-domain) test containing 5000 sentences for each sentence length up to 23 words (this corresponds to a maximal nesting depth of four RC), for a total of 75000 test sentences.

4.2 Models

We experimented with three strong neural latent constituency tree baselines: Neural PCFG and Compound PCFG (Kim et al., 2019a, henceforth, NPCFG and CPCFG) and Unsupervised Recurrent Neural Networks Grammars (URNNG, Kim et al., 2019b). Since the full parametrization of these three models would take too much space, we refer the interested reader to the original papers and recall only their most salient features.

NPCFG and CPCFG These models are based on a neural parametrization of PCFGs in Chomsky normal form. Both assume the number of nonterminal symbols of the grammar to be fixed as a hyperparameter. To generate a sentence, these models start from a designated nonterminal symbol S and recursively apply rewrite rules of the form $X \to \sigma$, replacing some nonterminal X with a sequence σ of one or two (terminal or nonterminal) symbols, until there remains only terminal symbols. The difference between NPCFG and CPCFG lies in how they model the choice of a rewrite rule. NPCFG parametrizes the probability of rewriting X with $X \to \sigma$ as proportional to $e^{f(u_X, v_\sigma)}$ where u_X and v_σ are learned embeddings for the left-hand side and right-hand side of the rule, and f is a neural network. CPCFG aims at weakening the context-free assumptions by making each step in the derivation dependent on a global context vector z. It thus generates z from a gaussian prior before applying the rewriting process. z is then shared between every rewriting decision, and the probability of $X \to \sigma$ becomes proportional to $e^{f(u_X \cdot z, v_\sigma)}$ where $\cdot$ is vector concatenation.

URNNG URNNG does not involve a discrete space of symbols and rules. It relies instead on a transition-based system inspired from shift-reduce parsers. A sentence is generated by successively applying SHIFT or REDUCE actions to a stack of

[6] It is hard to give a lower bound since PCFG minimization is undecidable.

tree fragments, until an end-of sentence symbol is generated. SHIFT generates a word and moves it on top of the stack, while REDUCE merges the two top elements of the stack into a single tree fragment. At every step, the model also maintains a stack of hidden states. The choice of the next action is parametrized as a function of the top-element of the stack of hidden states, and each action updates the stack of hidden states using a stack recurrent neural network (Kuncoro et al., 2017).

The tested models thus have important differences: URNNG, unlike the two others, generates words in a strict left-to-right order and does not involve symbolic rules. NPCFG is the only model whose decisions depend only on a **finite** set of configurations. CPCFG and URNNG stand on opposite sides of the parsing/language modeling tradeoff, with NPCFG and CPCFG achieving better parsing performance but much worse perplexity than URNNG on natural language. Unlike NPCFG, CPCFG and URNNG most likely have expressive capabilities beyond PCFG, though this has (to our knowledge) not be formally established. In particular, URNNG's use of stack LSTM suggests (Merrill et al., 2020; Weiss et al., 2018) that it could model data from the `+mce` conditions without branching structures, which lies beyond PCFG's strong expressive power[7]. Finally, the models use distinct strategies to marginalize over latent trees and estimate the probabilities of sentences: NPCFG and CPCFG use dynamic programs[8] whereas URNNG uses REINFORCE with control variate.

We tested the three models under the best set of hyperparameters[9] respectively reported in Kim et al. (2019a) and Kim et al. (2019b), using the authors' orginal implementation. In particular, we used CPCFG with 30 nonterminal symbols and 60 preterminal symbols, which means that, for the `-sr` configuration, a perfect language model and parser lies within the searched class of CPCFG models (cf §3.3), hence, that perfect parsing accuracy can be achieved on the test set.

We trained four instances of each model (CPCFG, NPCFG and URNNG) on a machine with a single RTX4090 GPU. Each run was initialized with a different random seed, on every training dataset. Training duration was controlled by the number of steps, to enable fair comparisons across datasets of different sizes. We trained for a maximum of 10^5 steps, performing validation every 500 steps, stopping early when perplexity on the development set failed to improve for three consecutive validations. Early stopping always triggered before the maximum number of steps was reached. Detailed hyperparameters are provided in Appendix A, and selected training, development and testing datasets, code and parsed test data are available at `https://github.com/suzuyuta/BriGap-2_2025`.

4.3 Metrics

We measured performances according to sentence-level (unlabeled) **F1 score**, the standard metrics for unsupervised constituency parsing. However, since our training data is designed around specific incentives for self-embedding of NP and CP, it does not reflect common linguistic arguments *e.g.* the positioning of the subject above VP, which are nevertheless evaluated by F1. We therefore report the recall on constituents of type NP and CP.

5 Results and discussion

Figure 2 presents the results achieved by the three models under the four configurations and four data sizes. The reported numbers are the mean performance and standard deviation across all four runs. Detailed results for each run are available in the appendix B (Table 3, 4, and 5).

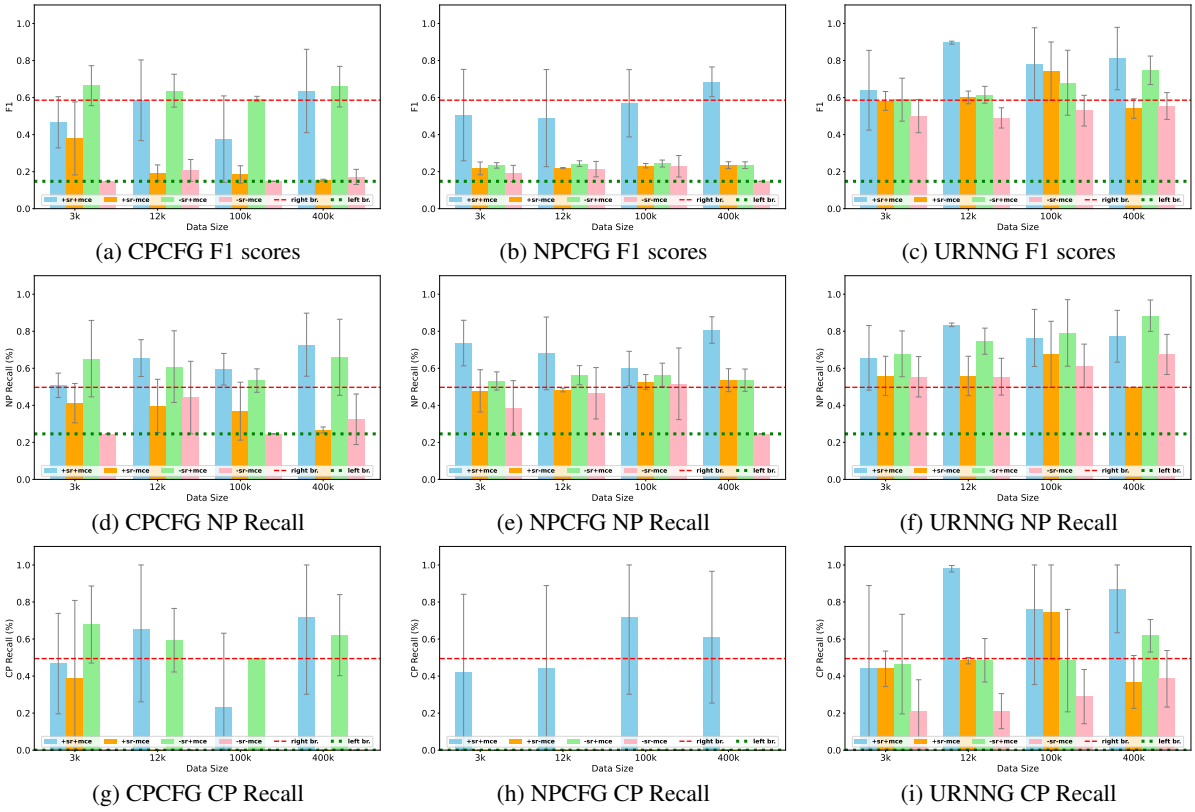
Overall impact of `sr` and `mce` The `+sr+mce` configuration (light blue) achieves the best results across the board, and the `-sr-mce` (pink) the worst results, confirming intuitions. The `-sr+mce` (light green) configuration tends to yield better result than the `+sr-mce` (orange), though there are exceptions for URNNG. This suggests that, to the extent that these phenomena are captured in our data, latent tree models are more sensitive to multiple center embedding than selectional restrictions. Furthermore, `+sr` configulations had the effect of increasing the number of preterminal symbols used in NPCFG and CPCFG models.

Model comparison URNNG seems to consistently achieve either comparable, or better performance than CPCFG, across all data sizes and configurations. This is rather surprising: for English, Kim et al. (2019a) reports a performance of 60.1

[7] URNNG essentially reduces to a standard LSTM language model when operating on linear tree structures.

[8] CPCFG uses variational inference to marginalize over z.

[9] Hyperparameters mainly involve embedding dimensions, hidden state dimensions, number of preterminals and nonterminals (when relevant), and number of KL annealing step for variational inference in URNNG.



(a) CPCFG F1 scores (b) NPCFG F1 scores (c) URNNG F1 scores

(d) CPCFG NP Recall (e) NPCFG NP Recall (f) URNNG NP Recall

(g) CPCFG CP Recall (h) NPCFG CP Recall (i) URNNG CP Recall

Figure 2: Sentence-level F1, NP and CP recall for CPCFG and URNNG. Dashed lines show the right-branching (red) and left-branching baseline scores (green)

for the best PCFG model against a performance of 52.6 for the best URNNG model on the Penn Treebank (Marcus et al., 1993). Hence, the (empirically) worse natural language parser achieved the better score on our artificial data. URNNG also provides the best language models across configurations, which is less surprising (perplexity scores estimated on training data are available in the appendix). Both models show a tendency to induce linear structures in the configuration `-sr-mce`, especially when less data is available. However, they seem to have opposite biases with CPCFG preferring **left-branching** structures, and URNNG **right-branching** ones.

Differences in `mce` impact `mce` strongly impacts all three models, but the impact on CPCFG is particularly dramatic, as no model ever beats the right-branching baseline in any of the `-mce` configuration. We find it remarkable, that despite plausibly able to express `mce` without branching structure (see § 4.2), the `-sr+mce` configuration pushes URNNG towards inducing NP constituents (though, not CP constituents).

Differences in `sr` impact The effect of selectional restrictions alone (`+sr-mce`) varies across models: CPCFG and NPCFG improve NP induction but not CPs, whereas URNNG shows the opposite pattern, especially excelling at CP recall with 100k data. Interestingly, NPCFG's F1 score drops sharply under `-sr`, though NP recall remains relatively stable. The contrast between `+sr+mce` and `-sr+mce` is more pronounced in NPCFG and URNNG than in CPCFG, probably because CPCFG cannot explicitly model selectional restrictions with its context vector z. Since z uniformely affects every occurence of a nonterminal symbol, lexical items generated from the same symbol share the same distribution, and CPCFG must assign different symbols to nouns with different semantic features. Consequently it has to encode semantic variation within a fixed inventory of 30 nonterminals and 60 preterminals. In contrast, URNNG may exploit its hidden state to model such distinctions more flexibly.

Performance correlations We used Spearman's rank correlation to assess the relationship between perplexity, training size, and syntactic performance (F1, NP/CP/VP recall; see Appendix C). CPCFG

and URNNG showed strong negative correlations between perplexity and performance, especially NP recall, suggesting that lower perplexity often aligns with better parsing. This trend was less clear for NPCFG. In contrast, correlations with training size were weaker. With small datasets, models typically saw the full data multiple times before converging, so seed-related variation mainly reflected data ordering. For larger datasets, perplexity often plateaued early, leading to convergence before full data exposure and greater sensitivity to the specific subset encountered.

Robustness to semantic and syntactic variation We further evaluated our models using test sets from the `-sr-mce` configuration and, for the models trained on `+mce`, `-sr+mce` configurations. Parsing performance remained consistent across these settings, suggesting that models do not rely on semantic cues, and that (when exposed to both kind of RCs) they learn to treat subject and object RC similarly. This indicates that self-embedding structures are either jointly acquired or jointly missed.

6 Limitations

The tested latent tree models obviously have very high variance under most configurations (URNNG on `+sr+mce` being an exception). Though the problem is pervasive in grammar induction, additional runs could help increase statistical significance. Second, comparison with more models would be very informative. In particular a comparison between the recent Tensor Decomposition PCFG model (Yang et al., 2021), since the former increased number of symbols could maybe overcome CPCFG apparently limitation to benefit from `sr`. Another limitation lies in the latent non/preterminal symbols in CPCFG and NPCFG, which we did not analyze in detail; future work is needed to better understand how these symbols relate to syntactic and semantic categories. Finally verbs and nouns are very unbalanced in our lexicon (more than in reality) and this asymmetry could have some effects, *e.g.* on some models' preferences for a given flow of information (from subject to verb vs. from verb to subject).

7 Conclusion

We have designed a controlled experimental setting to assess the respective effect of two linguistic phenomena (one categorical, multiple center embedding and one semantic, selectional restrictions) on latent tree models induction. Testing three well established latent tree baselines in these settings allows to make general observations on the relative strength of the two phenomena, and report differences in their impact on the tested models. While we focused on constituency models in this study, our methodology and data are readily applicable to the dependency setting, and testing dependency latent models is one of our future avenues of research.

Acknowledgments

We would like to thank anonymous reviewers for their suggestions and comments. This research was funded by the Natural Sciences and Engineering Research Council of Canada (RN001462).

References

C.L. Baker. 1995. *English Syntax*. MIT Press.

Yonatan Bisk and Julia Hockenmaier. 2013. An HDP model for inducing Combinatory Categorial Grammars. *Transactions of the Association for Computational Linguistics*, 1:75–88.

Glenn Carroll and Eugene Charniak. 1992. *Two experiments on learning probabilistic dependency grammars from corpora*. Department of Computer Science, Univ.

N. Chomsky. 1956. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124.

Morten H. Christiansen and Maryellen C. MacDonald. 2009. A usage-based approach to recursion in sentence processing. *Language Learning*, 59:126–161.

Alexander Clark. 2001. Unsupervised induction of stochastic context-free grammars using distributional clustering. In *Proceedings of the ACL 2001 Workshop on Computational Natural Language Learning (ConLL)*.

Jason Eisner and Giorgio Satta. 1999. Efficient parsing for bilexical context-free grammars and head automaton grammars. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 457–464, College Park, Maryland, USA. Association for Computational Linguistics.

Phu Mon Htut, Kyunghyun Cho, and Samuel Bowman. 2018. Grammar induction with neural language models: An unusual replication. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4998–5003, Brussels, Belgium. Association for Computational Linguistics.

Lifeng Jin, Finale Doshi-Velez, Timothy Miller, William Schuler, and Lane Schwartz. 2018. Unsupervised grammar induction with depth-bounded PCFG. *Transactions of the Association for Computational Linguistics*, 6:211–224.

Fred Karlson. 2007. Constraints on multiple center-embedding of clauses. *Journal of Linguistics*, 43(2):365–392.

Yoon Kim, Chris Dyer, and Alexander Rush. 2019a. Compound probabilistic context-free grammars for grammar induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2369–2385, Florence, Italy. Association for Computational Linguistics.

Yoon Kim, Alexander Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. 2019b. Unsupervised recurrent neural network grammars. 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 1105–1117, Minneapolis, Minnesota. Association for Computational Linguistics.

Dan Klein and Christopher Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 478–485, Barcelona, Spain.

Dan Klein and Christopher D Manning. 2001. Distributional phrase structure induction. In *Proceedings of the ACL 2001 Workshop on Computational Natural Language Learning (ConLL)*.

Dan Klein and Christopher D. Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 128–135, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A Smith. 2017. What do recurrent neural network grammars learn about syntax? In *EACL (1)*.

Nur Lan, Michal Geyer, Emmanuel Chemla, and Roni Katzir. 2022. Minimum description length recurrent neural networks. *Transactions of the Association for Computational Linguistics*, 10:785–799.

Guy Lapalme. 2020. The jsrealb text realizer: Organization and use cases. *arXiv preprint arXiv:2012.15425*.

K. Lari and S. J. Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4:35–56.

Bingzhi Li, Guillaume Wisniewski, and Benoît Crabbé. 2023. Assessing the capacity of transformer to abstract syntactic representations: A contrastive analysis based on long-distance agreement. *Transactions of the Association for Computational Linguistics*, 11:18–33.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

William Merrill, Gail Weiss, Yoav Goldberg, Roy Schwartz, Noah A. Smith, and Eran Yahav. 2020. A formal hierarchy of RNN architectures. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 443–459, Online. Association for Computational Linguistics.

Paul Molins and Guy Lapalme. 2015. Jsrealb: A bilingual text realizer for web programming. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pages 109–111.

Mark-Jan Nederhof and Giorgio Satta. 2011. Splittability of bilexical context-free grammars is undecidable. *Computational Linguistics*, 37(4):867–879.

Barbara H. Partee, Alice ter Meulen, and Robert E. Wall. 1990. *Mathematical Methods in Linguistics. Corrected first edition.* Kluwer Academic Publishers, Dordrecht.

John K Pate and Mark Johnson. 2016. Grammar induction from (lots of) words alone. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 23–32, Osaka, Japan. The COLING 2016 Organizing Committee.

Yikang Shen, Zhouhan Lin, Chin-wei Huang, and Aaron Courville. 2018. Neural language modeling by jointly learning syntax and lexicon. In *International Conference on Learning Representations*.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *International Conference on Learning Representations*.

C. Tellier. 2003. *Éléments de syntaxe du français: méthodes d'analyse en grammaire générative*. Morin.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the practical computational power of finite precision RNNs for language recognition. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 740–745, Melbourne, Australia. Association for Computational Linguistics.

Songlin Yang, Yanpeng Zhao, and Kewei Tu. 2021. PCFGs can do better: Inducing probabilistic context-free grammars with many symbols. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1487–1498, Online. Association for Computational Linguistics.

Shunyu Yao, Binghui Peng, Christos Papadimitriou, and Karthik Narasimhan. 2021. Self-attention networks can process bounded hierarchical languages. 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 3770–3785, Online. Association for Computational Linguistics.

Hao Zhu, Yonatan Bisk, and Graham Neubig. 2020. The return of lexical dependencies: Neural lexicalized pcfgs. *Transactions of the Association for Computational Linguistics*, 8:647–661.

A Hyperparameters

The hyperparameters used in our experiments largely follow those reported in prior work (Kim et al., 2019a,b). We conducted four runs for each model (CPCFG, NPCFG, and URNNG) with the following randomly selected seeds: `3435`, `648708704`, `1320159950`, and `603135965`. Table 2 summarizes the hyperparameter settings. In the experiments with the NPCFG, the `z_dim` parameter of the CPCFG was set to 0.

Table 2: Hyperparameters

CPCFG (NPCFG)

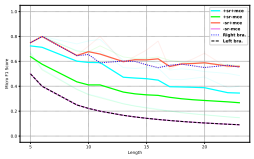
Description	Value	Flag
latent dimension	64	`-z_dim`
number of preterminal states	60	`-t_states`
number of nonterminal states	30	`-nt_states`
symbol embedding dimension	256	`-state_dim`
hidden dim for variational LSTM	512	`-h_dim`
embedding dim for variational LSTM	512	`-w_dim`
starting learning rate	0.001	`-lr`
gradient clipping	3	`-max_grad_norm`
max sentence length cutoff start	30	`-max_length`
increment max length each epoch	1	`-len_incr`
final max length cutoff	40	`-final_max_length`
Adam β_1	0.75	`-beta1`
Adam β_2	0.999	`-beta2`
which GPU to use	0	`-gpu`
validation every N steps	3000	`-val_every`
increment max length every N steps	3000	`-incr_step`
early stopping patience (epochs)	5	`-early_stopping_patience`
minimum training steps	10000	`-min_steps`

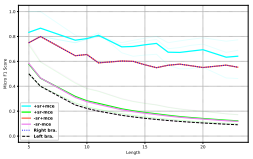
URNNG

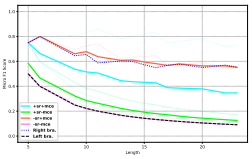
Description	Value	Flag
hidden dim (LM/RNNG)	650	`-w_dim`
hidden dim (LM/RNNG)	650	`-h_dim`
hidden dim (variational RNN)	256	`-q_dim`
number of layers (LM & stack LSTM)	2	`-num_layers`
dropout rate	0.5	`-dropout`
include EOS in val PPL (0/1)	0	`-count_eos_ppl`
no LR decay before this	8	`-min_epochs`
IWAE samples (eval)	5	`-mc_samples`
samples for score-function grads	8	`-samples`
starting learning rate	1	`-lr`
LR for inference network q	0.0001	`-q_lr`
LR for action layer	0.1	`-action_lr`
LR decay factor	0.5	`-decay`
KL warmup steps	10000	`-kl_warmup`
steps to train q	10000	`-train_q_steps`
uniform init range	0.1	`-param_init`
grad clipping (model)	5	`-max_grad_norm`
grad clipping (q)	1	`-q_max_grad_norm`
validation every N steps	3000	`-val_every`
minimum training steps	1500	`-min_steps`

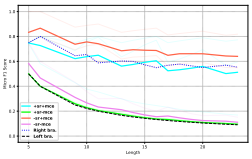
B All scores by length and All Results tables

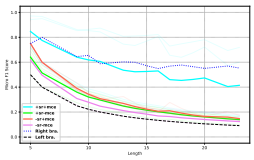
F1 score

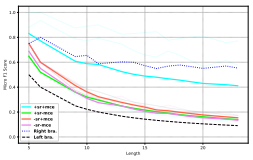

CPCFG F1 3k

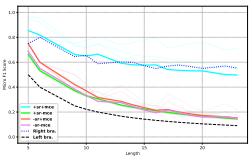

CPCFG F1 12k

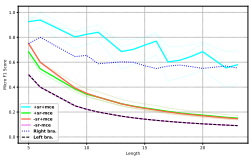

CPCFG F1 100k

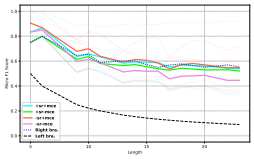

CPCFG F1 400k

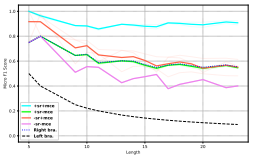

NPCFG F1 3k

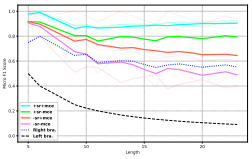

NPCFG F1 12k

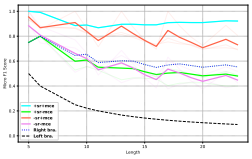

NPCFG F1 100k


NPCFG F1 400k

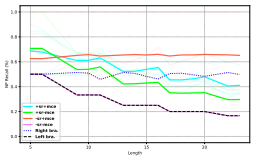

URNNG F1 3k

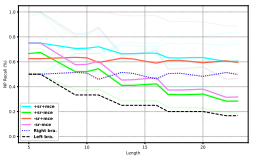

URNNG F1 12k

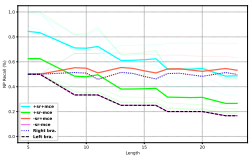

URNNG F1 100k

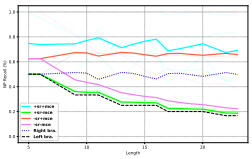

URNNG F1 400k

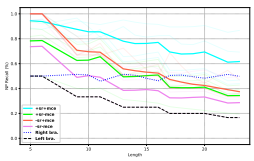
NP prediction recall

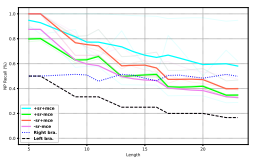

CPCFG NP 3k

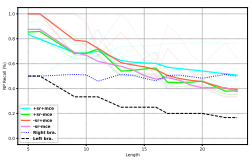

CPCFG NP 12k

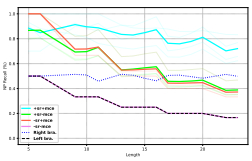

CPCFG NP 100k

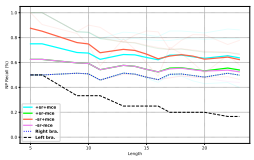

CPCFG NP 400k

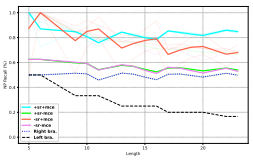

NPCFG NP 3k

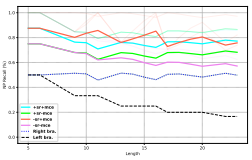

NPCFG NP 12k

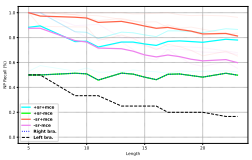

NPCFG NP 100k


NPCFG NP 400k

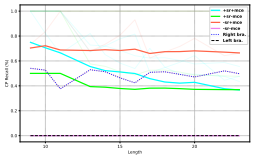

URNNG NP 3k


URNNG NP 12k

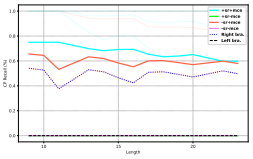

URNNG NP 100k


URNNG NP 400k

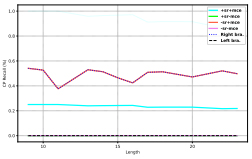
CP prediction recall



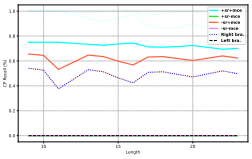
CPCFG CP 3k



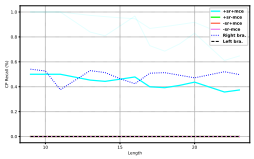
CPCFG CP 12k



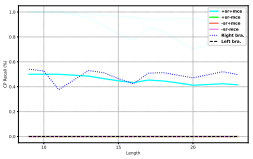
CPCFG CP 100k



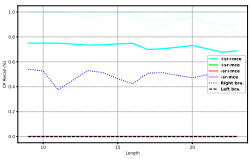
CPCFG CP 400k



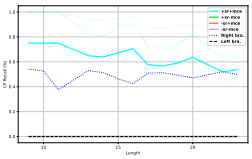
NPCFG CP 3k



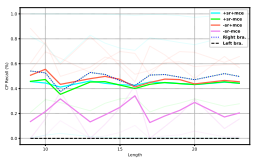
NPCFG CP 12k



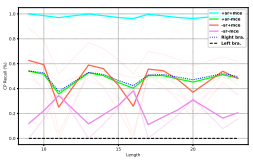
NPCFG CP 100k



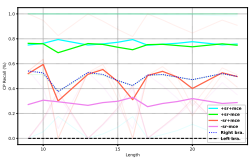
NPCFG CP 400k



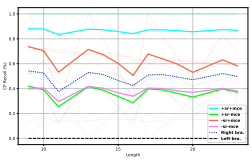
URNNG CP 3k



URNNG CP 12k

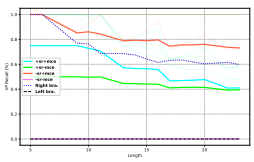


URNNG CP 100k

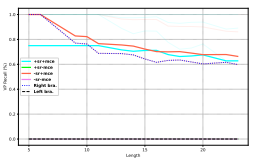


URNNG CP 400k

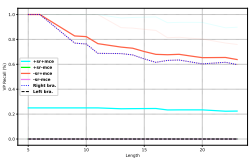
VP prediction recall



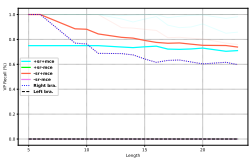
CPCFG VP 3k



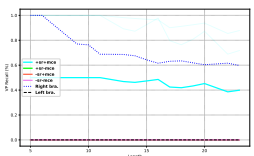
CPCFG VP 12k



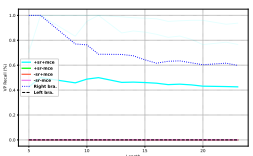
CPCFG VP 100k



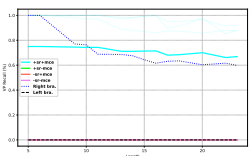
CPCFG VP 400k



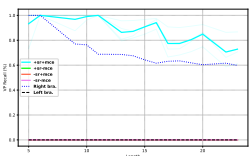
NPCFG VP 3k



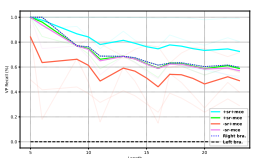
NPCFG VP 12k



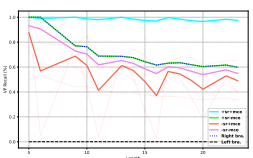
NPCFG VP 100k



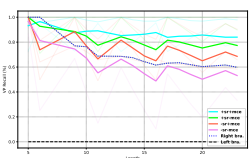
NPCFG VP 400k



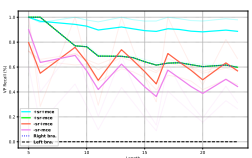
URNNG VP 3k



URNNG VP 12k



URNNG VP 100k



URNNG VP 400k

Table 3: CPCFG Results

Data Type	Size	Seed	Val PPL	Last Step	**F1**	**NP**	**CP**	**VP**	**nb NT**	**nb PreT**
+sr+mce	3k	0	24.93	9000	0.51	0.42	0.58	0.65	10	30
		1	24.41	10000	0.58	0.6	0.66	0.72	10	28
		2	23.43	11000	0.23	0.52	0.0	0.0	10	33
		3	22.03	9000	0.54	0.48	0.63	0.75	14	32
	12k	0	20.63	13000	0.73	0.71	0.92	0.95	16	30
		1	18.97	12000	0.61	0.66	0.7	0.8	9	22
		2	20.18	9500	0.77	0.75	1.0	1.0	10	23
		3	21.74	8000	0.22	0.49	0.0	0.0	11	33
	100k	0	20.11	11500	0.78	0.72	0.92	0.95	12	28
		1	22.14	10500	0.26	0.63	0.0	0.0	11	25
		2	22.13	8000	0.22	0.49	0.0	0.0	11	30
		3	20.43	12500	0.24	0.54	0.0	0.0	7	24
	400k	0	20.17	13500	0.67	0.54	0.89	0.93	11	32
		1	18.87	11000	0.77	0.75	1.0	1.0	11	25
		2	19.29	13000	0.26	0.62	0.0	0.0	10	28
		3	18.5	14000	0.84	0.99	0.99	0.99	8	29
+sr-mce	3k	0	18.54	8500	0.62	0.39	1.0	1.0	13	28
		1	21.56	7000	0.15	0.25	0.0	0.0	7	21
		2	19.4	9000	0.23	0.51	0.0	0.0	6	26
		3	18.34	9000	0.52	0.5	0.56	0.73	7	30
	12k	0	19.79	11000	0.16	0.27	0.0	0.0	8	32
		1	16.6	15500	0.26	0.63	0.0	0.0	8	34
		2	19.36	10000	0.16	0.28	0.0	0.0	7	29
		3	18.64	10500	0.2	0.41	0.0	0.0	7	25
	100k	0	19.71	8500	0.17	0.3	0.0	0.0	6	24
		1	16.43	17500	0.27	0.64	0.0	0.0	7	38
		2	19.38	10500	0.15	0.27	0.0	0.0	7	29
		3	20.05	9500	0.15	0.26	0.0	0.0	8	25
	400k	0	19.17	7000	0.15	0.25	0.0	0.0	7	22
		1	20.01	6500	0.15	0.26	0.0	0.0	7	22
		2	19.22	10000	0.16	0.29	0.0	0.0	7	30
		3	19.75	8500	0.16	0.28	0.0	0.0	6	26
-sr+mce	3k	0	39.23	10000	0.64	0.61	0.72	0.81	13	29
		1	47.51	7000	0.59	0.5	0.49	0.66	7	22
		2	46.74	6500	0.59	0.5	0.49	0.66	7	18
		3	34.58	11500	0.85	1.0	1.0	1.0	9	23
	12k	0	43.03	8000	0.59	0.5	0.49	0.66	7	16
		1	45.7	6500	0.59	0.5	0.49	0.66	6	16
		2	44.68	6500	0.59	0.5	0.49	0.66	6	15
		3	35.25	10000	0.79	0.94	0.89	0.93	7	14
	100k	0	44.93	8000	0.59	0.5	0.49	0.66	7	14
		1	41.64	7000	0.62	0.64	0.49	0.85	11	12
		2	44.21	8000	0.59	0.5	0.49	0.66	8	16
		3	44.6	7500	0.59	0.5	0.49	0.66	6	12
	400k	0	34.49	14000	0.62	0.64	0.49	0.85	12	25
		1	32.42	12000	0.85	1.0	1.0	1.0	9	20
		2	44.95	7000	0.59	0.5	0.49	0.66	7	12
		3	44.75	8000	0.59	0.5	0.49	0.66	8	18
-sr-mce	3k	0	42.17	6500	0.15	0.25	0.0	0.0	6	19
		1	37.25	7000	0.15	0.25	0.0	0.0	7	22
		2	42.7	6500	0.15	0.25	0.0	0.0	6	16
		3	42.31	6500	0.15	0.25	0.0	0.0	6	18
	12k	0	28.14	13000	0.27	0.64	0.0	0.0	7	17
		1	39.5	7000	0.15	0.25	0.0	0.0	6	23
		2	38.91	6500	0.15	0.25	0.0	0.0	6	18
		3	32.06	14500	0.27	0.64	0.0	0.0	7	19
	100k	0	38.56	8000	0.15	0.25	0.0	0.0	6	27
		1	38.83	6500	0.15	0.25	0.0	0.0	6	22
		2	38.87	7000	0.15	0.25	0.0	0.0	6	23
		3	34.74	6500	0.15	0.25	0.0	0.0	6	18
	400k	0	38.56	7500	0.15	0.25	0.0	0.0	6	25
		1	38.72	8000	0.15	0.25	0.0	0.0	7	21
		2	32.36	9500	0.24	0.56	0.0	0.0	7	21
		3	34.44	8000	0.15	0.25	0.0	0.0	7	22

Table 4: NPCFG Results

Data Type	Size	Seed	Val PPL	Last Step	**F1**	**NP**	**CP**	**VP**	**nb NT**	**nb PreT**
+sr+mce	3k	0	20.67	9000	0.72	0.81	0.77	0.85	9	28
		1	25.58	6500	0.25	0.6	0.0	0.0	9	23
		2	25.48	6500	0.27	0.64	0.0	0.0	7	25
		3	23.28	8500	0.78	0.9	0.9	0.94	11	25
	12k	0	25.98	7000	0.2	0.42	0.0	0.0	7	22
		1	21.91	10000	0.82	0.97	0.95	0.97	7	23
		2	24.42	7500	0.27	0.64	0.0	0.0	7	29
		3	21.01	10500	0.67	0.69	0.82	0.83	10	24
	100k	0	22.71	10000	0.27	0.64	0.0	0.0	8	23
		1	20.36	9000	0.62	0.48	1.0	0.89	12	22
		2	20.51	11500	0.64	0.55	0.92	0.94	10	18
		3	25.2	8000	0.75	0.73	0.95	0.96	8	22
	400k	0	21.74	12500	0.69	0.71	0.77	0.85	10	34
		1	20.65	9000	0.77	0.87	0.9	0.93	11	26
		2	21.19	9500	0.73	0.88	0.77	0.81	10	20
		3	20.25	14000	0.56	0.76	0.0	0.77	14	33
+sr-mce	3k	0	22.32	9500	0.22	0.5	0.0	0.0	8	26
		1	19.56	9500	0.25	0.58	0.0	0.0	8	28
		2	19.99	8500	0.24	0.54	0.0	0.0	8	23
		3	22.52	6500	0.16	0.29	0.0	0.0	7	28
	12k	0	21.5	8500	0.21	0.47	0.0	0.0	8	27
		1	18.72	9500	0.22	0.49	0.0	0.0	8	24
		2	18.6	12000	0.22	0.49	0.0	0.0	12	27
		3	18.95	10000	0.22	0.49	0.0	0.0	12	24
	100k	0	18.65	10500	0.22	0.49	0.0	0.0	8	26
		1	18.81	8500	0.24	0.54	0.0	0.0	8	31
		2	18.84	12000	0.22	0.49	0.0	0.0	11	24
		3	21.26	11000	0.25	0.58	0.0	0.0	8	26
	400k	0	18.83	7500	0.22	0.49	0.0	0.0	8	21
		1	18.63	12000	0.22	0.48	0.0	0.0	9	31
		2	18.72	11000	0.27	0.64	0.0	0.0	8	22
		3	19.45	8500	0.23	0.53	0.0	0.0	8	24
-sr+mce	3k	0	36.07	9500	0.22	0.49	0.0	0.0	8	17
		1	31.45	10000	0.23	0.52	0.0	0.0	8	16
		2	38.44	8000	0.26	0.61	0.0	0.0	12	20
		3	36.33	10000	0.22	0.5	0.0	0.0	6	16
	12k	0	43.74	8000	0.27	0.64	0.0	0.0	6	13
		1	35.22	10000	0.22	0.49	0.0	0.0	6	17
		2	48.96	7500	0.24	0.57	0.0	0.0	6	18
		3	34.2	9500	0.24	0.56	0.0	0.0	9	18
	100k	0	35.9	10500	0.27	0.66	0.0	0.0	13	13
		1	35.26	10000	0.22	0.5	0.0	0.0	8	16
		2	43.02	8000	0.23	0.53	0.0	0.0	8	15
		3	40.46	7000	0.24	0.57	0.0	0.0	8	19
	400k	0	40.92	8000	0.27	0.64	0.0	0.0	7	12
		1	35.78	8000	0.22	0.49	0.0	0.0	7	16
		2	43.69	7000	0.22	0.49	0.0	0.0	9	16
		3	33.54	11000	0.23	0.52	0.0	0.0	8	14
-sr-mce	3k	0	27.2	8000	0.25	0.59	0.0	0.0	10	20
		1	39.07	7000	0.15	0.25	0.0	0.0	6	19
		2	31.22	10000	0.21	0.46	0.0	0.0	8	14
		3	35.33	6500	0.15	0.25	0.0	0.0	5	10
	12k	0	27.04	8000	0.26	0.63	0.0	0.0	7	15
		1	32.63	8000	0.22	0.49	0.0	0.0	11	15
		2	34.11	6500	0.15	0.25	0.0	0.0	5	14
		3	26.57	9000	0.22	0.49	0.0	0.0	10	16
	100k	0	27.12	10000	0.31	0.79	0.0	0.0	8	13
		1	33.06	6500	0.23	0.54	0.0	0.0	7	13
		2	34.46	6500	0.15	0.25	0.0	0.0	8	12
		3	26.79	7500	0.22	0.49	0.0	0.0	9	15
	400k	0	33.5	7000	0.15	0.25	0.0	0.0	6	18
		1	33.13	7500	0.15	0.25	0.0	0.0	6	19
		2	33.73	8000	0.15	0.25	0.0	0.0	6	25
		3	33.98	8000	0.15	0.25	0.0	0.0	7	12

Table 5: URNNG Results

Data Type	Size	Seed	Val PPL	Last Step	**F1**	**NP**	**CP**	**VP**
+sr+mce	3k	0	15.96	3500	0.43	0.5	0.0	0.66
		1	14.25	4500	0.91	0.85	0.99	0.99
		2	16.24	3500	0.42	0.47	0.01	0.63
		3	14.53	4500	0.79	0.81	0.77	0.82
	12k	0	11.78	8500	0.91	0.84	1.0	1.0
		1	12.45	6000	0.9	0.84	0.99	0.97
		2	12.64	6000	0.89	0.83	0.95	0.97
		3	11.84	8500	0.89	0.82	0.98	0.99
	100k	0	12.54	6000	0.87	0.85	0.99	0.82
		1	11.69	8000	0.91	0.85	1.0	0.98
		2	11.47	9000	0.91	0.85	1.0	0.98
		3	12.48	7000	0.44	0.5	0.06	0.66
	400k	0	11.71	7500	0.91	0.85	1.0	0.98
		1	11.57	8500	0.91	0.85	1.0	0.98
		2	11.87	7500	0.91	0.85	1.0	0.98
		3	12.2	8500	0.52	0.53	0.46	0.67
+sr-mce	3k	0	13.27	5500	0.59	0.5	0.49	0.66
		1	13.32	4500	0.51	0.5	0.27	0.66
		2	13.58	3500	0.58	0.5	0.49	0.66
		3	12.86	5500	0.65	0.74	0.49	0.62
	12k	0	11.59	6000	0.57	0.5	0.45	0.66
		1	11.34	6000	0.58	0.5	0.49	0.66
		2	11.75	5500	0.58	0.5	0.49	0.66
		3	10.88	6500	0.66	0.74	0.49	0.66
	100k	0	10.28	9000	0.91	0.85	1.0	1.0
		1	10.71	8000	0.59	0.5	0.49	0.66
		2	10.37	8000	0.89	0.85	1.0	0.89
		3	10.91	7000	0.59	0.5	0.49	0.66
	400k	0	10.84	7500	0.59	0.5	0.49	0.66
		1	11.09	7000	0.46	0.5	0.15	0.66
		2	11.53	6000	0.53	0.5	0.34	0.66
		3	10.83	7500	0.59	0.5	0.49	0.66
-sr+mce	3k	0	42.06	3500	0.43	0.49	0.0	0.64
		1	42.11	4000	0.75	0.82	0.63	0.74
		2	39.79	4500	0.62	0.75	0.59	0.42
		3	50.4	4500	0.56	0.66	0.63	0.34
	12k	0	37.71	5000	0.67	0.76	0.52	0.63
		1	35.82	6000	0.54	0.63	0.34	0.68
		2	35.84	6000	0.62	0.81	0.42	0.42
		3	35.61	6000	0.63	0.79	0.66	0.37
	100k	0	35.52	7000	0.95	0.98	0.92	0.92
		1	35.54	7000	0.67	0.82	0.4	0.64
		2	35.37	7000	0.46	0.5	0.16	0.66
		3	35.45	6500	0.64	0.86	0.45	0.7
	400k	0	35.38	8000	0.84	0.99	0.72	0.69
		1	35.32	9000	0.76	0.91	0.63	0.61
		2	35.36	8000	0.76	0.89	0.64	0.63
		3	35.42	7500	0.63	0.75	0.48	0.52
-sr-mce	3k	0	44.85	3500	0.46	0.5	0.16	0.66
		1	35.63	3500	0.43	0.48	0.03	0.64
		2	44.61	3500	0.65	0.74	0.49	0.61
		3	36.76	5500	0.46	0.5	0.16	0.66
	12k	0	33.37	6500	0.46	0.5	0.16	0.66
		1	43.12	5000	0.46	0.5	0.16	0.66
		2	33.41	6000	0.46	0.5	0.16	0.66
		3	36.63	5000	0.58	0.73	0.37	0.46
	100k	0	33.33	6000	0.46	0.5	0.15	0.66
		1	43.01	5000	0.66	0.74	0.49	0.66
		2	35.19	5000	0.54	0.72	0.36	0.38
		3	33.2	7500	0.46	0.5	0.16	0.66
	400k	0	35.73	5000	0.46	0.5	0.16	0.66
		1	33.25	8500	0.57	0.78	0.55	0.28
		2	33.21	7500	0.53	0.68	0.34	0.41
		3	38.0	6500	0.66	0.74	0.49	0.66

C Spearman's Test Results

Table 6: Spearman's Test Results: CPCFG

Data Type	Pair	Spearman's ρ	p-value	Significance
+sr+mce	PPL-Size	-0.7276	0.0014	**
	F1-Size	0.2795	0.2944	n.s.
	F1-PPL	-0.6013	0.0137	*
	NP-Size	0.4983	0.0495	*
	NP-PPL	-0.7119	0.002	**
	CP-Size	0.1933	0.4732	n.s.
	CP-PPL	-0.5535	0.0261	*
	VP-Size	0.1933	0.4732	n.s.
	VP-PPL	-0.5656	0.0224	*
+sr-mce	PPL-Size	0.1576	0.5598	n.s.
	F1-Size	-0.5209	0.0385	*
	F1-PPL	-0.728	0.0014	**
	NP-Size	-0.3588	0.1723	n.s.
	NP-PPL	-0.733	0.0012	**
	CP-Size	-0.506	0.0455	*
	CP-PPL	-0.4065	0.1182	n.s.
	VP-Size	-0.506	0.0455	*
	VP-PPL	-0.4065	0.1182	n.s.
-sr+mce	PPL-Size	-0.2183	0.4167	n.s.
	F1-Size	-0.0489	0.8574	n.s.
	F1-PPL	-0.8415	0.0	***
	NP-Size	0.0209	0.9386	n.s.
	NP-PPL	-0.8483	0.0	***
	CP-Size	-0.2153	0.4231	n.s.
	CP-PPL	-0.6712	0.0044	**
	VP-Size	0.0209	0.9386	n.s.
	VP-PPL	-0.8483	0.0	***
-sr-mce	PPL-Size	-0.4672	0.068	n.s.
	F1-Size	0.0356	0.8958	n.s.
	F1-PPL	-0.6811	0.0037	**
	NP-Size	0.0356	0.8958	n.s.
	NP-PPL	-0.6811	0.0037	**
	CP-Size	NaN	NaN	n.s.
	CP-PPL	NaN	NaN	n.s.
	VP-Size	NaN	NaN	n.s.
	VP-PPL	NaN	NaN	n.s.

Table 7: Spearman's Test Results: NPCFG

Data Type	Pair	Spearman's ρ	p-value	Significance
+sr+mce	PPL-Size	-0.5457	0.0288	*
	F1-Size	0.1946	0.4702	n.s.
	F1-PPL	-0.3392	0.1987	n.s.
	NP-Size	0.1216	0.6536	n.s.
	NP-PPL	-0.2094	0.4363	n.s.
	CP-Size	0.1814	0.5014	n.s.
	CP-PPL	-0.4111	0.1137	n.s.
	VP-Size	0.1788	0.5077	n.s.
	VP-PPL	-0.3902	0.1351	n.s.
+sr-mce	PPL-Size	-0.5461	0.0286	*
	F1-Size	0.2271	0.3977	n.s.
	F1-PPL	-0.111	0.6823	n.s.
	NP-Size	0.0873	0.7479	n.s.
	NP-PPL	0.031	0.9092	n.s.
	CP-Size	NaN	NaN	n.s.
	CP-PPL	NaN	NaN	n.s.
	VP-Size	NaN	NaN	n.s.
	VP-PPL	NaN	NaN	n.s.
-sr+mce	PPL-Size	0.097	0.7208	n.s.
	F1-Size	0.0189	0.9447	n.s.
	F1-PPL	0.314	0.2363	n.s.
	NP-Size	-0.0061	0.982	n.s.
	NP-PPL	0.3403	0.1972	n.s.
	CP-Size	NaN	NaN	n.s.
	CP-PPL	NaN	NaN	n.s.
	VP-Size	NaN	NaN	n.s.
	VP-PPL	NaN	NaN	n.s.
-sr-mce	PPL-Size	0.0243	0.929	n.s.
	F1-Size	-0.2534	0.3436	n.s.
	F1-PPL	-0.8164	0.0001	***
	NP-Size	-0.2534	0.3436	n.s.
	NP-PPL	-0.8164	0.0001	***
	CP-Size	NaN	NaN	n.s.
	CP-PPL	NaN	NaN	n.s.
	VP-Size	NaN	NaN	n.s.
	VP-PPL	NaN	NaN	n.s.

Table 8: Spearman's Test Results: URNNG

Data Type	Pair	Spearman's ρ	p-value	Significance
+sr+mce	PPL-Size	-0.7155	0.0018	**
	F1-Size	0.4054	0.1193	n.s.
	F1-PPL	-0.7405	0.001	**
	NP-Size	0.4818	0.0588	n.s.
	NP-PPL	-0.6258	0.0095	**
	CP-Size	0.5371	0.0319	*
	CP-PPL	-0.8149	0.0001	***
	VP-Size	0.1543	0.5682	n.s.
	VP-PPL	-0.6229	0.01	**
+sr-mce	PPL-Size	-0.7155	0.0018	**
	F1-Size	0.0	1.0	n.s.
	F1-PPL	-0.5779	0.019	*
	NP-Size	-0.0799	0.7688	n.s.
	NP-PPL	-0.4339	0.0931	n.s.
	CP-Size	-0.0349	0.898	n.s.
	CP-PPL	-0.5142	0.0416	*
	VP-Size	0.3202	0.2266	n.s.
	VP-PPL	-0.5911	0.0159	*
-sr+mce	PPL-Size	-0.9459	0.0	***
	F1-Size	0.5595	0.0242	*
	F1-PPL	-0.4381	0.0897	n.s.
	NP-Size	0.5769	0.0193	*
	NP-PPL	-0.4875	0.0554	n.s.
	CP-Size	0.2433	0.364	n.s.
	CP-PPL	-0.174	0.5192	n.s.
	VP-Size	0.2005	0.4565	n.s.
	VP-PPL	-0.1975	0.4635	n.s.
-sr-mce	PPL-Size	-0.4972	0.0501	n.s.
	F1-Size	0.3824	0.1438	n.s.
	F1-PPL	0.2374	0.376	n.s.
	NP-Size	0.4289	0.0974	n.s.
	NP-PPL	0.1261	0.6417	n.s.
	CP-Size	0.381	0.1454	n.s.
	CP-PPL	0.1756	0.5154	n.s.
	VP-Size	-0.1602	0.5533	n.s.
	VP-PPL	0.2585	0.3336	n.s.

D Statistics of Non-/Pre-terminal Symbols for CPCFG and NPCFG

Table 9: Mean number of Non-/Pre-terminal symbols

Model	Data Type	Mean nb NT	(std)	Mean nb PreT	(std)
CPCFG	+sr+mce	10.69	(2.05)	28.25	(3.44)
	+sr-mce	7.38	(1.58)	27.56	(4.49)
	-sr+mce	8.12	(2.09)	17.62	(4.83)
	-sr-mce	6.38	(0.48)	20.69	(2.93)
NPCFG	+sr+mce	9.38	(1.96)	24.81	(4.22)
	+sr-mce	8.69	(1.49)	25.75	(2.8)
	-sr+mce	8.06	(1.95)	16.0	(2.15)
	-sr-mce	7.44	(1.77)	15.62	(3.66)

Plural Interpretive Biases: A Comparison Between Human Language Processing and Language Models

Jia Ren
University of Massachusetts Amherst
jiaren@umass.edu

Abstract

Human communication routinely relies on plural predication, and plural sentences are often ambiguous (see, e.g., Scha, 1984; Dalrymple et al., 1998a, to name a few). Building on extensive theoretical and experimental work in linguistics and philosophy, we ask whether large language models (LLMs) exhibit the same interpretive biases that humans show when resolving plural ambiguity. We focus on two lexical factors: (i) the *collective bias* of certain predicates (e.g., size/shape adjectives) and (ii) the *symmetry bias* of predicates. To probe these tendencies, we apply two complementary methods to premise–hypothesis pairs: an embedding-based heuristic using OpenAI's `text-embedding-3-large/small` (OpenAI, 2024, 2025) with cosine similarity, and supervised NLI models (`bart-large-mnli`, `roberta-large-mnli`) (Lewis et al., 2020; Liu et al., 2019; Williams et al., 2018a; Facebook AI, 2024b,a) that yield asymmetric, calibrated entailment probabilities. Results show partial sensitivity to predicate-level distinctions, but neither method reproduces the robust human pattern, where neutral predicates favor entailment and strongly non-symmetric predicates disfavor it. These findings highlight both the potential and the limits of current LLMs: as cognitive models, they fall short of capturing human-like interpretive biases; as engineering systems, their representations of plural semantics remain unstable for tasks requiring precise entailment.

1 Introduction

Plural sentences permit multiple readings. For example, *the boys lifted the table* allows a *collective* reading (acting together) or a *distributive* reading (each acted separately). Even without context, human listeners show robust preferences, making plurality a rich testbed for evaluating whether language models track the same interpretive pressures.

We ask two questions. As *cognitive models*, do LLMs exhibit human-like interpretive biases in out-of-the-blue contexts? As *engineering systems*, do they represent plural semantics robustly enough for tasks requiring precise entailment? If such biases emerge, they may be encoded in linguistic distributions; if not, it shows the limits of text-only training.

We focus on two tendencies: the *collective bias*, where predicates vary in supporting collective over distributive readings, and the *symmetry bias*, where reciprocals differ in favoring symmetric interpretations. To probe these, we apply two methods to premise–hypothesis pairs: (i) cosine similarity with OpenAI's `text-embedding-3-large/small`, a simple but symmetric and uncalibrated proxy for entailment, and (ii) NLI models (`bart-large-mnli`, `roberta-large-mnli`), which provide asymmetric, probabilistic entailment judgments. We treat both as entailment-strength signals and compare their agreement and fit to human data.

2 Background

2.1 Collective Bias

Pluralities like *the students*, *John and Mary* are widely used in natural language. However, the semantics and pragmatics of predicating properties on plural entities is a complex issue. The complexity comes from the ambiguity of plural predications (Beck and Sauerland, 2000; Beck, 2001; Landman, 1989a,b; Link, 1983; Scha, 1984; Schwarzschild, 1996).

For example, for a sentence *the boys lifted the table*, a plural entity *the boys* is involved. The sentence allows for various interpretations. The first and most intuitive reading of the sentence is that all the boys lifted the table together. The property of table lifting applies to the plural entity *the boys* as a whole. This is commonly referred to as *the*

Proceedings of the Second Workshop on the Bridges and Gaps between Formal and Computational Linguistics (BriGap-2), pages 97–105

collective reading of plural predications. Another possible reading of the sentence is that the boys each lifted the table. The property of table lifting applies to each atom of the plural entity *the boys*. This is commonly referred to as *the distributive reading* of plural predications. In addition to the collective and distributive readings, there are also many intermediate readings. For example, the sentence is also true in a scenario where the boys were separated into groups, each group of boys lifted the table together.

Plural predication sentences are inherently ambiguous. However, this ambiguity does not hinder the efficiency or effectiveness of human communication. Rather than causing confusion, certain interpretation is usually prominent.

Experimental work has shown that collective readings are generally easier to access than distributive readings (Frazier et al., 1999; Dotlačil and Brasoveanu, 2021). However, this preference is not uniform: many special cases reveal a weak collective bias. For instance, Dotlačil and Brasoveanu (2021) find that the preference for collective interpretations disappears in cases of lexical distributivity, where the distributive meaning is encoded directly in the predicate. A well-documented example arises with adjectives of size and shape, which strongly promote distributive interpretations (Quine, 1960; Schwarzschild, 2011; Scontras and Goodman, 2017; Syrett, 2015; Maldonado, 2012; Zhang, 2013). Syrett (2015) show that this bias emerges early, in children as young as three. For example, the adjective *large* strongly favors a distributive reading: when interpreting *the boxes are big*, the most natural construal is that each box is big. By contrast, predicates such as *heavy* allow both collective and distributive interpretations: *the boxes are heavy* may mean that each box is heavy, or that the boxes are heavy as a group, even if no individual box is particularly heavy.

Why the preference arises remains an open question in the literature. One line of explanation attributes the interpretive bias to lexical semantics, certain predicates are argued to be semantically incompatible with collective readings due to their scalar or gradable nature, as in the case of size adjectives like *big* or *tall* (Schwarzschild, 2011; Maldonado, 2012; Zhang, 2013). Another line of research suggests that the preference is shaped by pragmatic reasoning or contextual factors; for instance, when interpreting size predicates, comparisons are naturally drawn at the level of each object. However, when the discourse context is set up appropriately, collective readings can emerge even for predicates that are otherwise known to strongly favor distributive interpretations (Scontras and Goodman, 2017). Scontras and Goodman (2017) collect natural-occurring examples of plural predications from the British National Corpus. For frequent plural sentences, Scontras and Goodman (2017) tests people's judgment of the salient interpretations of these sentences. The authors also manipulate the contexts of the same plural sentences and show that contexts can influence how salient the distributive reading is, thus refuting the lexical views mentioned above.

2.2 Symmetric Bias

The second generalization is that certain predicates evoke a symmetric bias, making the salient interpretations of plural sentences stronger compared to those without a symmetric bias (Beck, 2001; Dalrymple et al., 1998b; Gleitman et al., 1996; Poortman et al., 2018). For example, for the sentence *John, Mary and Bill knew each other*, the most salient reading is that John knew Mary, Mary knew John, John knew Bill, Bill knew John, Mary knew Bill and Bill knew Mary. In other words, every person knew every other person. The salient interpretation is symmetric between the atoms of the individual. For the sentence *John, Mary and Bill were hitting each other*, the most salient reading is not as strong as the one described just now. The most salient reading is that every person was either hitting or was being hit by some other person. The reading is weaker than the reading for the *knew* sentence. The reading is not symmetric between atoms of the plural.

To explain the difference in the strength of salient readings, many works focus on the use of contextual and lexical information for the selection between different readings (Dalrymple et al., 1998b; Sabato and Winter, 2012; Mari, 2013). In an influential paper, Dalrymple et al. (1998b) introduces a principle named the *Strongest Meaning Hypothesis*. The principle predicts that the strongest possible interpretation will be salient in case of ambiguity. According to the principle, the predicate *hit* was more non-symmetric to *know*. In a hitting event, a person most likely either hit someone or was hit by someone, but not both. In a knowing event, a person can both know someone and be known by someone. Thus, in the reciprocal sentences mentioned above, *John, Mary and Bill knew*

each other has a reading which is stronger than *John, Mary and Bill were hitting each other*.

Some predicates exhibit a higher degree of symmetry than others, a generalization supported by various strands of empirical and theoretical work. Gleitman et al. (1996) observes that predicates like *be similar* are interpreted more symmetrically than predicates like *love* or *help*, suggesting that conceptual representations influence perceived symmetry. Winter (2018) provides a formal semantic account of such variability, arguing that reciprocal alternations reflect systematic differences in predicate symmetry, with certain predicates favoring reciprocal interpretations more naturally. Complementing these perspectives, Poortman et al. (2018) propose the Maximal Typicality Hypothesis, showing through experimental evidence that the interpretation of reciprocal expressions depends on how typically symmetrical a predicate is perceived to be, with more symmetrical predicates leading to stronger reciprocal inferences.

More specifically, Poortman et al. (2018) investigates how verb concepts influence the interpretation of plural reciprocal sentences in Dutch and Hebrew. Building on prior work, they first examine Hebrew data to evaluate the Strongest Meaning Hypothesis (SMH). Contrary to this prediction, their results show that participants often opt for weaker interpretations. Poortman et al. (2018) argue that this pattern reflects the sensitivity of reciprocal quantification to the underlying verb concept, and propose the Maximal Typicality Hypothesis. According to the hypothesis, a reciprocal sentence is most acceptable in a "core situation", one that is both maximally extensive and maximally typical for the verb concept—and may also be acceptable in supersets of that situation, but not in others. They conducted two experiments in Dutch, one typicality ranking task assessing symmetry preferences across different verbs, and a truth-value judgment task with plural sentences using those verbs. The findings reveal systematic variation in how many patients are typically associated with each agent across verb types, and this variation significantly affects reciprocal interpretation. The stronger the verb's bias toward non-symmetric scenarios, the more likely participants are to adopt a weaker reciprocal reading.

Collectively, these studies support the view that symmetry is a graded and conceptually grounded property of predicates, with consequences for both interpretation and grammatical alternation.

2.3 Goals

Prior experimental work has not examined how LLMs resolve these plural ambiguities. Using the two-method framework above (embedding similarity vs. NLI probabilities), we ask whether model signals reflect human biases for distributivity and symmetry. Our main questions are:

1. In out-of-the-blue contexts, do model-based entailment signals reflect the human *collective bias*?

2. In out-of-the-blue contexts, do model-based entailment signals reflect the human *symmetry bias*?

3. Do the two methods agree—cosine similarity vs. NLI $p(\text{entailment})$—on which readings are preferred, and where do they diverge?

We operationalize these questions by applying cosine similarity with OpenAI embeddings and by estimating $p(\text{entailment} \mid P, H)$ with `bart-large-mnli` and `roberta-large-mnli`, then comparing model outputs with prior human data (Scontras and Goodman, 2017; Poortman et al., 2018).

3 Data Collection

3.1 Collective Bias

To test the collective bias, we use the same dataset from Scontras and Goodman (2017). The authors selected the 40 most frequent combinations of the form "the nouns were adjective" from the British National Corpus, ensuring ecological validity by using naturally occurring language patterns. Participants were asked to judge what each sentence meant on a slider bar, with one end representing the paraphrase "the nouns each were adjective" (distributive interpretation) and the other end representing "the nouns together were adjective" (collective interpretation). This dataset is particularly valuable for our purposes because it provides a systematic comparison of human interpretive preferences across a range of predicate types, allowing us to assess whether language models capture the same semantic distinctions that guide human comprehension. The full list of sentences is provided in the appendix.

Participants showed a wide range of ratings across the 40 sentences, as shown in Figure 1. The figure displays the collective endorsement rate with

95% confidence intervals for each sentence tested. The x-axis shows the 40 sentences, and the y-axis indicates the proportion of responses toward the collective end of the slider bar. The results reveal systematic variation in how strongly participants favored collective versus distributive interpretations, with some sentences (e.g., "results disappointing") showing high collective endorsement and others (e.g., "classes small") showing low collective endorsement.

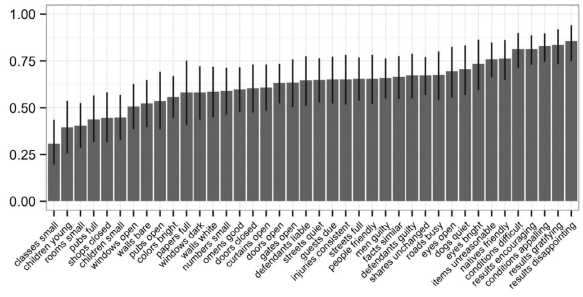


Figure 1: Results of Scontras and Goodman (2017). Collective endorsement rates with 95% confidence intervals for 40 sentences of the form "the nouns were adjective." Higher values indicate stronger preference for collective interpretations.

In our experiments, we adapt these same sentences as input to large language models to examine whether similar interpretive biases emerge in model predictions.

3.2 Symmetric Bias

To test the symmetric bias, we use the same dataset from Poortman et al. (2018). In the original paper, the authors tested 18 Dutch verbs among Dutch speakers. The 18 verbs are categorized into three types based on their patient preference.

- Type 1 (neutral): envy, know, understand, admire, miss, hate
- Type 2 (non-symmetric-preference): pinch, hit, caress, stab, shoot, grab
- Type 3 (strong non-symmetric-preference): kiss, dress, kick, lash out, bite, lick

Each verb was embedded in a sentence of the form "A, B and C Verb each other", where A, B, and C were random proper names. Participants were asked to perform truth value judgment tasks for these sentences under two types of scenarios: one depicting a symmetric action and the other a non-symmetric action. From a generalized linear mixed model (GLMM) logistic regression analysis, it was observed that in the symmetric scenarios, sentences with neutral verbs were rated significantly better than non-symmetric verbs, and non-symmetric verbs were rated as significantly better than strongly non-symmetric verbs. In the non-symmetric scenarios, the reverse pattern was observed.

In our experiments, we adapt these same sentences as input to large language models to examine whether similar interpretive biases emerge in model predictions.

4 Experiment 1

In Experiment 1, we ask whether an embedding-based metric recovers two human tendencies in plural interpretation: the collective–distributive preference and symmetry effects. We compute cosine similarity between bare and explicitly marked paraphrases using sentence embeddings from OpenAI's `text-embedding-3-large` and `text-embedding-3-small`.

4.1 Method

Experiment 1a: collective bias In this experiment, we examine the semantic similarity between two types of plural sentences: (i) bare plural sentences (Sentence 1), which lack explicit distributive or collective markers, and (ii) marked plural sentences (Sentence 2), which contain overt markers indicating distributive or collective interpretations. Examples of the tested sentences are as below.

1. Sentence 1: *the classes were small.*
2. Sentence 2 (distributive): *the classes each were small.*
3. Sentence 2 (collective): *the classes together were small.*

We use OpenAI's `text-embedding-3-large` and `text-embedding-3-small` to compute sentence embeddings and evaluate how similarly the two sentence types are represented.

Experiment 1b: symmetric bias In this experiment, we examine the semantic similarity between two types of plural sentences: (i) bare plural reciprocal sentences (Sentence 1), which lack explicit symmetric markers, and (ii) marked symmetric sentences (Sentence 2), which contain overt markers indicating symmetric interpretations. Examples of the tested sentences are as below.

1. Sentence 1: *the children knew each other.*
2. Sentence 2 (symmetric): *every child knew every other child.*

We use OpenAI's `text-embedding-3-large` and `text-embedding-3-small` to compute sentence embeddings and evaluate how similarly the two sentence types are represented.

We embed two sentences as vectors $\mathbf{u}, \mathbf{v}$ in the same semantic space and compute their cosine similarity,

$$\cos(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \, \|\mathbf{v}\|} \in [-1, 1].$$

The embeddings for each pair of sentences are passed into a cosine similarity function, which returns a similarity score that is first linearly rescaled to [0, 1], then passed through a sigmoid transformation to smooth the scale. Cosine similarity is a measure used to quantify how similar two vectors are, regardless of their magnitude. Because cosine is scale-invariant and bounded, it is a convenient, single-number proxy for semantic relatedness. Some recent discussions on the application of the methods can be found in Steck et al. (2024) and You (2025), to name a few. It calculates the cosine of the angle between the two vectors in a multi-dimensional space, which reflects their orientation rather than their length. The resulting value ranges from –1 to 1, where 1 indicates that the vectors are pointing in the same direction (i.e., they are very similar), 0 means they are orthogonal (i.e., unrelated), and –1 means they are diametrically opposed. In natural language processing and information retrieval, cosine similarity is commonly used to compare text documents represented as word or sentence embeddings, allowing for efficient comparison of semantic content.

Our stimuli come in minimally different paraphrase sets that make the target interpretation explicit. For each bare sentence (e.g., *the classes were small*), we compare its similarity to a distributive paraphrase (*the classes each were small*) versus a collective paraphrase (*the classes together were small*). If a model encodes the *collective bias* that humans show for size/shape predicates, the bare sentence should be closer (higher cosine) to the distributive paraphrase than to the collective one. Analogously, for reciprocals, we compare a bare reciprocal (e.g., *A, B and C knew each other*) to stronger, fully symmetric paraphrases versus weaker, non-symmetric paraphrases. If the model encodes a *symmetry bias*, the bare reciprocal should sit closer to the fully symmetric paraphrase. Cosine similarity thus provides a simple, model-agnostic diagnostic that turns these preferences into ranked distances.

We use OpenAI's dedicated embedding models rather than hidden states from general-purpose LMs for three practical reasons. (i) Task fit: these models are trained explicitly to produce sentence embeddings whose geometry reflects semantic similarity, making cosine a meaningful signal out of the box. (iii) Sensitivity analysis: using two sizes (`*-large` and `*-small`) lets us check whether conclusions depend on embedding capacity: convergent patterns across sizes increase confidence that findings are not an artifact of a single representation. We still analyze limitations below: cosine is symmetric ($\cos(P, H) = \cos(H, P)$) and uncalibrated, so it cannot by itself model directional entailment—hence our complementary NLI experiment.

4.2 Result

We compared similarity scores across model conditions using paired t-tests, Wilcoxon signed-rank tests, and OLS regressions with item fixed effects. These analyses test whether mean differences between conditions are reliably different from zero while accounting for within-item variation. The results show that for the large model, distributive sentences were judged slightly more similar to their base forms than collective sentences (Mean Diff $\approx 0.01, p < .05$), whereas the small model showed no significant distributive–collective difference. In contrast, collective scores from the small model were systematically higher than those from the large model (Mean Diff ≈ 0.03, $p < 10^{-10}$), a large and robust effect. Overall, the large model appears sensitive to subtle distributive–collective contrasts, confirms the similarity between the bare sentences and their distributive/collective marked counterparts.

Figure3 ranks the cosine similarity scores of collective sentences in the large model from low to high. It serves as a language model analogue to Figure1. Comparing Figure1 and Figure3, we see that although both humans and the language model display a gradient of bias, the specific patterns of bias are not the same. The x-axis, which corresponds to item numbers, highlights the relative ranking of sentences, and this ranking for humans differs substantially from that of the language model. This

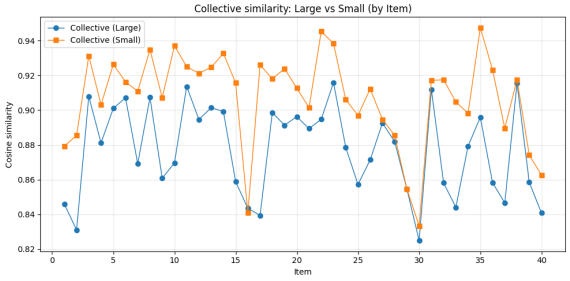


Figure 2: Comparison of similarity scores of the collective condition across the large and small embedding models.

indicates that while the large language model is sensitive to collective bias, its behavior diverges markedly from human judgments.

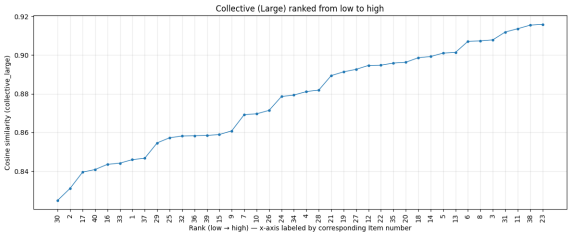


Figure 3: Comparison of similarity scores between collective and distributive conditions in the large embedding model.

Figure 4 presents the average scores of the large and small models across sentences with three verb types: neutral, non-symmetric, and strongly non-symmetric. The large model shows a steady rise across the three types, with the highest average score for strongly non-symmetric items. This indicates that the more non-symmetric a verb is, the greater the similarity between the original sentence and its symmetric paraphrase. However, this pattern contradicts the results observed in human experiments. By contrast, the small model exhibits a flatter trend, showing only minimal improvement between neutral and non-symmetric cases and even a slight decrease for strongly non-symmetric items—again diverging from human results. Overall, these findings demonstrate a clear difference between human judgments and model behavior.

Interim summary In this section, we used a cosine similarity task to test collective and symmetric biases in OpenAI's `text-embedding-3-large/small`. The large model shows slight collective–distributive contrasts, but overall neither model reproduces the collective bias observed in humans.

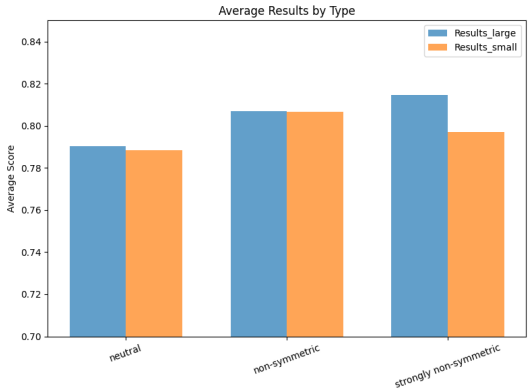


Figure 4: Comparison of average similarity scores for each verb type across the large and the small models.

5 Experiment 2

Whereas Experiment 1 employed a cosine similarity task with sentence embeddings, which measures the degree of semantic closeness between original sentences and their variants, Experiment 2 relies on supervised NLI models that explicitly calculate entailment probabilities between a premise and a hypothesis. In this way, the two experiments complement each other: cosine similarity offers an indirect, gradient measure of interpretive bias, while NLI provides a direct, categorical assessment of whether one interpretation is supported by another. Each pair consisted of a base sentence (premise) and a variant reflecting a collective/ symmetric interpretation (hypothesis). To evaluate whether the hypothesis was entailed by the premise, we employed two supervised NLI models: `bart-large-mnli` and `roberta-large-mnli`, both fine-tuned on the Multi-Genre Natural Language Inference (MNLI) corpus (Williams et al., 2018b). The results show that, unlike human participants, language models do not show the collective/symmetric bias observed in human language use.

5.1 Method

We complement the cosine similarity experiments with a second paradigm based on supervised Natural Language Inference (NLI) models, specifically `bart-large-mnli` and `roberta-large-mnli`. In this setup, we calculate entailment probabilities between the same sentence pairs tested in Experiment 1. Namely, sentences and their corresponding collective, distributive or symmetric paraphrases. Whereas cosine similarity captures geometric closeness in embedding space without reference to spe-

cific inference relations, the NLI framework explicitly asks whether one sentence (the premise) entails another (the hypothesis). This difference is crucial: cosine similarity measures general semantic similarity, while NLI probes whether the model recognizes logical inference patterns such as symmetry.

The logic of the NLI experiments is as follows. We take each sentence as the premise and the collective/distributive or symmetric paraphrase as the hypothesis, and then use the supervised NLI models to compute the probability that the hypothesis is entailed. If a model assigns high entailment probability to the symmetric hypothesis, this suggests that it encodes a collective/distributive or symmetric bias for that predicate. Thus, these experiments go beyond the embedding-based cosine similarity approach by directly testing whether models treat symmetric interpretations as logically following from collective descriptions. Together, the two approaches provide complementary perspectives: cosine similarity reveals gradient semantic affinities, while NLI directly assesses whether symmetric readings are licensed as inferences.

We select `bart-large-mnli` and `roberta-large-mnli` because both are strong, widely used supervised NLI models that have been fine-tuned on the Multi-Genre Natural Language Inference (MNLI) dataset, which covers a broad range of sentence types and inference relations. `roberta-large-mnli` represents a transformer model trained with a robust masked-language-modeling objective, while `bart-large-mnli` combines an encoder–decoder architecture with denoising pretraining, making it particularly effective for classification tasks like NLI. Using these complementary models allows us to test whether our findings hold across different architectures, ensuring that observed patterns are not idiosyncratic to a single model design.

5.2 Result

Figure 5 reveal strikingly different patterns between the two models. For BART, there is a strong positive correlation between collective and distributive entailment probabilities ($r = 0.713$): items that score higher in the collective condition also tend to score higher in the distributive condition. The scores are the probability assigned to entailment, i.e., how strongly the model believes the collective/distributive interpretation logically follows from the original sentence. The regression line lies close to the 45-degree reference, suggesting that BART treats the two conditions as related and often raises both probabilities together. By contrast, for RoBERTa, the relationship is essentially flat ($r = 0.061$). The regression slope is close to zero, and the points are scattered broadly around the vertical axis, indicating that collective scores have little predictive value for distributive scores. This divergence suggests that BART encodes a stronger link between collective and distributive interpretations, whereas RoBERTa treats them as largely independent dimensions. Moreover, RoBERTa strongly favors the distributive interpretations, while BART, in contrast, treats the two interpretations as positively correlated without systematically preferring one over the other.

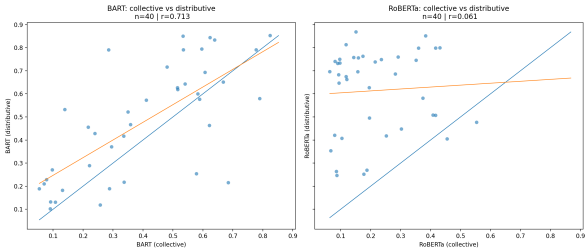


Figure 5: Comparison of the probabilities for collective/distributive pairs of BART and RoBERTa.

Figure 5 presents the RoBERTa counterpart to Figures 1 and 3, showing the collective scores ranked from low to high. While the model exhibits a gradient of scores, the ranking of items diverges considerably from the human rankings reported in Scontras and Goodman (2017). Notably, the pattern resembles the results obtained from the cosine similarity task, suggesting that the two approaches capture similar model-internal preferences rather than human-like biases.

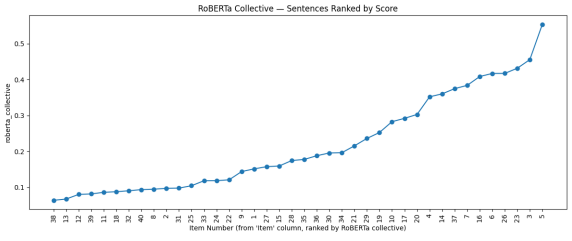


Figure 6: Sentences ranked by their RoBERTa collective scores.

Figure 7 compares the average symmetric scores of BART and RoBERTa across three verb types: neutral, non-symmetric, and strongly non-symmetric. BART consistently assigns high sym-

metric scores across all categories, with means ranging from 0.78 to 0.90, suggesting a strong tendency to treat symmetric paraphrases as entailed regardless of verb type. RoBERTa, in contrast, yields substantially lower scores (around 0.33–0.41), but exhibits a clearer distinction between categories: symmetric scores rise for strongly non-symmetric verbs relative to neutral and non-symmetric verbs. Again, as in the cosine similarity tasks, the human inference results are not shown here, which predict that neutral verbs should have the highest probability of entailment, while strongly non-symmetric verbs should have the lowest. BART flattens the differences almost entirely, while RoBERTa reverses the expected trend.

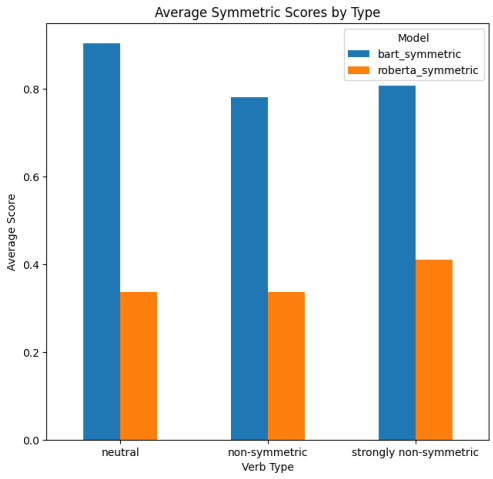


Figure 7: Average symmetric scores by verb types.

Interim summary In this section, we used an NLI task to test collective and symmetric biases. The models show slight sensitivity to plural-interpretation contrasts, but their behavior remains very different from human processing. Overall, the results are similar to those from the cosine similarity task, with neither method reproducing the robust biases observed in human judgments.

6 Limitations

Cosine similarity and NLI are imperfect proxies for human interpretation—the former symmetric and uncalibrated, the latter shaped by model-specific biases—so our results show alignment with human judgments but not competence. Stimulus design was limited to single agent–patient pairs (e.g., the boys with the table), reducing contextual variability to which human interpretation is sensitive. More broadly, by abstracting from discourse, world knowledge, and prosody, our study offers only coarse approximations. Future work should use richer behavioral methods and newer open-source models to assess alignment more fully.

7 Conclusion

This study asked whether large language models exhibit the same interpretive pressures that guide human comprehension of plural sentences. Using cosine similarity and NLI models, we probed collective and symmetry biases in out-of-the-blue contexts. The results show partial sensitivity to predicate-level distinctions, but neither method reproduced the robust human pattern—neutral verbs favoring entailment and strongly non-symmetric verbs disfavoring it. As cognitive models, LLMs therefore fall short of capturing human-like biases; as engineering systems, their representations of plural semantics remain unstable for tasks requiring precise entailment. These findings mark the limits of text-only training and point to future work in which we plan to incorporate visual cues, alongside richer context and more nuanced evaluation metrics, to better align model semantics with human judgments.

References

Sigrid Beck. 2001. Reciprocals Are Definites. *Natural Language Semantics*, 9(1):69–138. Publisher: Springer.

Sigrid Beck and Uli Sauerland. 2000. Cumulation is needed. *Natural Language Semantics*, 8(4):349–371.

Mary Dalrymple, Makoto Kanazawa, Yookyung Kim, Sam Mchombo, and Stanley Peters. 1998a. Reciprocal expressions and the concept of reciprocity. *Linguistics and Philosophy*, 21(2):159–210.

Mary Dalrymple, Makoto Kanazawa, Yookyung Kim, Sam Mchombo, and Stanley Peters. 1998b. Reciprocal Expressions and the Concept of Reciprocity.

Jakub Dotlačil and Adrian Brasoveanu. 2021. The representation and processing of distributivity and collectivity: ambiguity vs. underspecification. *Glossa: a journal of general linguistics*, 6(1):1–22.

Facebook AI. 2024a. `FacebookAI/roberta-large-mnli` — model card. `https://huggingface.co/FacebookAI/roberta-large-mnli`. RoBERTa fine-tuned on MultiNLI.

Facebook AI. 2024b. `facebook/bart-large-mnli` — model card. `https://huggingface.co/`

facebook/bart-large-mnli. BART fine-tuned on MultiNLI.

Lyn Frazier, Jeremy M. Pacht, and Keith Rayner. 1999. Taking on semantic commitments, ii: Collective versus distributive readings. *Cognition*, 70(1):87–104.

Lila R Gleitman, Henry Gleitman, Carol Miller, and Ruth Ostrin. 1996. Similar, and similar concepts.

Fred Landman. 1989a. Groups i. *Linguistics and Philosophy*, 12(6):559–605.

Fred Landman. 1989b. Groups ii. *Linguistics and Philosophy*, 12(6):723–744.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of ACL*.

Godehard Link. 1983. The Logical Analysis of Plurals and Mass Terms: A Lattice-theoretical Approach. page 22.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Violeta Vázquez Rojas Maldonado. 2012. *The Syntax and Semantics of Purépecha Noun Phrases and the Mass/Count Distinction*. Ph.d. dissertation, New York University, New York, NY.

Alda Mari. 2013. Each other, asymmetry and reasonable futures. *Journal of Semantics*, 31(2):209–261.

OpenAI. 2024. New embedding models and api updates. https://openai.com/index/new-embedding-models-and-api-updates/. Introduces `text-embedding-3-large` and `text-embedding-3-small`.

OpenAI. 2025. Vector embeddings – openai api documentation. https://platform.openai.com/docs/guides/embeddings. Model pages: `text-embedding-3-large`, `text-embedding-3-small`.

Eva B. Poortman, Marijn E. Struiksma, Nir Kerem, Naama Friedmann, and Yoad Winter. 2018. Reciprocal expressions and the Maximal Typicality Hypothesis. *Glossa: a journal of general linguistics*, 3(1).

Willard Van Orman Quine. 1960. *Word and Object*. MIT Press, Cambridge, MA.

Sivan Sabato and Yoad Winter. 2012. Relational domains and the interpretation of reciprocals. *Linguistics and Philosophy*, 35(3):191–241.

Remko J.H. Scha. 1984. *Distributive, Collective and Cumulative Quantification*, pages 131–158. De Gruyter Mouton, Berlin, Boston.

Roger Schwarzschild. 1996. *Pluralities*. Studies in Linguistics and Philosophy. Springer Netherlands.

Roger Schwarzschild. 2011. Stubborn distributivity, multiparticipant nouns and the count/mass distinction. In *Proceedings of the 39th Annual Meeting of the North East Linguistic Society (NELS 39)*, pages 661–678, Amherst, MA. GLSA, University of Massachusetts.

Gregory Scontras and Noah D. Goodman. 2017. Resolving uncertainty in plural predication. *Cognition*, 168:294–311.

Harald Steck, Chaitanya Ekanadham, and Nathan Kallus. 2024. Is Cosine-Similarity of Embeddings Really About Similarity? *arXiv preprint arXiv:2403.05440*. Analysis showing cosine similarities can be arbitrary in linear embedding models.

Kristen Syrett. 2015. Mapping properties to individuals in language acquisition. In *Proceedings of the 39th Annual Boston University Conference on Language Development (BUCLD 39)*, Somerville, MA. Cascadilla Press.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018a. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018b. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Yoad Winter. 2018. Symmetric predicates and the semantics of reciprocal alternations. *Semantics and Pragmatics*, 11(1):1.

Kisung You. 2025. Semantics at an Angle: When Cosine Similarity Works Until It Doesn't. *arXiv preprint arXiv:2504.16318*. Informal conceptual and empirical survey of cosine similarity limitations and alternatives.

Niina Ning Zhang. 2013. *Classifier Structures in Mandarin Chinese*. Mouton de Gruyter, Berlin.

Appendix

All plots, experimental data, and analysis scripts used in this paper are openly available at the following repository: https://github.com/ziaren/plurals-human-lm

Author Index

Bekki, Daisuke, 1, 15
Bernard, Timothée, 52
Brown, Jason, 8, 20

Crabbé, Benoit, 52

Daido, Hinari, 1

Iimura, Aoi, 15

Kallmeyer, Laura, 52
Kruschwitz, Kascha, 35

Matsubara, Mai, 1
Mielczarek, Lukas, 52
Mizuno, Teruyuki, 15

Patejuk, Agnieszka, 26
Pistotti, Timothy, 8, 20
Prasanth, , 72
Przepiórkowski, Adam, 26

Ren, Jia, 97

Spalek, Katharina, 52
Suzuki, Yutaka, 79

Tomita, Asa, 1

Venant, Antoine, 79

Witbrock, Michael J., 8, 20

Zodl, Paul, 35
Zymla, Mark-Matthias, 35