Preface: SCiL 2025 Editors' Note

Carolyn Jane Anderson
Wellesley College

Fred Mailhot
Dialpad, Inc.

Grusha Prasad
Colgate University

This volume contains research presented at the eighth annual meeting of the Society for Computation in Linguistics (SCiL), held in Eugene, Oregon, July 18-20, 2025.

Research was submitted to be reviewed either in the form of a paper, or as an abstract. The oral presentations, or talks, at the conference included both papers and abstracts. Authors of accepted abstracts were given the option of publishing an extended version; these are included with the papers in this volume.

In total, we received 75 main conference submissions, of which 20 were selected for oral presentation (~26%) and 29 for poster presentation (~38%). We also received 4 submissions for methodology lightning talks; 2 were accepted (~50%).

SCiL 2025 included a special Symposium on Computational Pragmatics. 6 talks were part of this Symposium, as well as many posters. The Symposium also included invited talks by Malihe Alikhani (Northeastern University), Daniel Fried (Carnegie Mellon University), Junyi Jessy Li (University of Texas at Austin), and Hannah Rohde (University of Edinburgh).

Further information can be found at our website: wellesley-easel-lab.github.io/SCiL2025/index.html.

We thank our reviewers for their indispensable help in selecting the research for presentation at the conference:

Bonnie Webber, Shira Wein, Christopher Potts, Timothee Mickus, Sebastian Schuster, Mai Al-Khatib, Clara Meister, Hayley Ross, Katrin Erk, Dylan Bumford, Philippe de Groote, Canaan Breiss, Aniello De Santo, Robert Malouf, Dongsung Kim, Alan C. L. Yu, Brian Dillon, Eric Raimy, Edward P. Stabler, Caleb Belth, Colin Wilson, Hossep Dolatian, Itamar Kastner, Giorgio Magri, Ollie Sayeed, Olga Zamaraeva, Thomas Graf, Joe Pater, Brandon Prickett, Gaja Jarosz, Jeffrey Heinz, Caitlin Smith, Andrew Lamont, Kasia Hitczenko, Adam Jardine, Jonathan Brennan, Christo Kirov, Tal Linzen, Nur Lan, Robert Frank, Lindy Comstock, Connor Mayer, Laurel Perkins, Suhas Arehalli, Tamar Johnson, Qihui Xu, Cassandra L Jacobs, Emily Morgan, Tim Hunter, Tiago Pimentel, Tracy Holloway King, Sheng-Fu Wang, Richard Futrell, Barend Beekhuizen, and Kyle Gorman.

Thanks also to Joe Pater for his help with publishing the proceedings.

We gratefully acknowledge Google as a corporate sponsor of SCiL 2025. We are also grateful to the U.S. National Science Foundation for partially funding the Symposium on Computational Pragmatics (BCS-2439863).

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 1-1.
Eugene, Oregon, July 18-20, 2025

Integrating Neural and Symbolic Components in a Model of Pragmatic Question-Answering

Polina Tsvilodub
University of Tübingen
first.last@uni-tuebingen.de

Robert D. Hawkins
Stanford University
hawkrobe@gmail.com

Michael Franke
University of Tübingen
first.last@uni-tuebingen.de

Abstract

Computational models of pragmatic language use have traditionally relied on hand-specified sets of utterances and meanings, limiting their applicability to real-world language use. We propose a neuro-symbolic framework that enhances probabilistic cognitive models by integrating LLM-based modules to propose and evaluate key components in natural language, eliminating the need for manual specification. Through a classic case study of pragmatic question-answering, we systematically examine various approaches to incorporating neural modules into the cognitive model—from evaluating utilities and literal semantics to generating alternative utterances and goals. We find that hybrid models can match or exceed the performance of traditional probabilistic models in predicting human answer patterns. However, the success of the neuro-symbolic model depends critically on how LLMs are integrated: while they are particularly effective for proposing alternatives and transforming abstract goals into utilities, they face challenges with truth-conditional semantic evaluation. This work charts a path toward more flexible and scalable models of pragmatic language use while illuminating crucial design considerations for balancing neural and symbolic components.

1 Introduction

Imagine you are a barista in a café with only three items in stock: iced coffee, soda, and Chardonnay. If a customer asks: “Do you have iced tea?”, you might naturally respond “I’m sorry, we don’t have iced tea, but I can make you an iced coffee!”. This situation exemplifies *pragmatic question answering*, where answerers commonly go beyond the literal question being asked (Clark, 1979). Classical accounts of the semantic meaning of questions and answers (e.g., Hamblin, 1973; Groenendijk and Stokhof, 1984; Hakulinen, 2001), maintain that polar questions like “Do you have iced tea?” are fully resolved by a polar answer {yes, no}. Yet humans routinely provide a *relevant* selection of additional information (e.g., mentioning the iced coffee, but not the Chardonnay).

Understanding what, exactly, makes an answer relevant has been a central question in the field of pragmatics, with extensive work investigating the contextual factors that shape answer selection (e.g. van Rooy, 2003; Stevens et al., 2016; Rothe et al., 2017). One recent framework for modeling these pragmatic choices is the Rational Speech Act framework (Frank and Goodman, 2012; Degen, 2023), which has been successfully applied to both question and answer selection (Hawkins et al., 2015; Hawkins and Goodman, 2017; Hawkins et al., to appear). The probabilistic cognitive models (PCMs) developed within this framework offer significant advantages through their transparent, explicit task decomposition and systematic error analysis (Farrell and Lewandowsky, 2018).

However, these models are typically limited to a small set of predefined examples, restricting their applicability to real-world scenarios. In contrast, Large Language Models (LLMs) offer a complementary set of capabilities. They can process open-ended natural language input and generate flexible responses, but often struggle with subtle pragmatic patterns (Hu et al., 2023; Ruis et al., 2023; Tsvilodub et al., 2024b) and lack the degree of explainability that makes PCMs so valuable for cognitive modeling (Zhao et al., 2023).

To address these complementary strengths and limitations, we explore a family of *neuro-symbolic* models, with different combinations of both approaches to leverage their respective strengths and to overcome known shortcomings.[1] Our ap-

[1] We use the term *neuro-symbolic* in the sense of a model that has *neural* network components (here, LLMs), that are scaffolded by a *symbolic* task analysis, i.e., integrated in a particular computational procedure. Other senses of the term also exist (Bhuyan et al., 2024).

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 2-17.
Eugene, Oregon, July 18-20, 2025

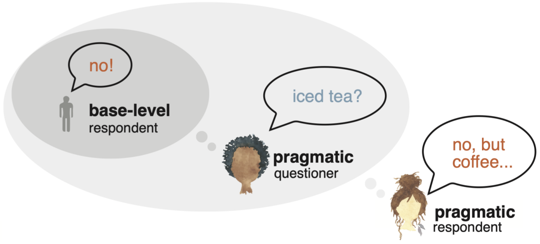


Figure 1: Probabilistic cognitive model (PCM) of pragmatic question answering. The PCM, built in the Rational Speech Act framework, implements recursive back-and-forth reasoning of rational agents. The questioner chooses a question based on their decision problem and an expectation of responses that any question might provoke. The respondent chooses a relevant response based on the decision problem inferred from the question.

proach builds on the task analysis developed in previous work on pragmatic question-answering (Hawkins et al., 2015; Hawkins and Goodman, 2017; Hawkins et al., to appear) in two ways. First, we use it as a *scaffolding structure* that determines the computational steps, with LLMs executing specific subtasks that would traditionally require manual specification in a PCM (Sections 3.2–3.3). Second, we verbalize (parts of) the scaffolding structure in a single prompt, relying on a single LLM call to solve the respective computational task (Section 3.4). This dual approach enables us to systematically investigate the tradeoffs between fine-grained task decomposition and end-to-end neural processing.

Our key contributions are as follows:

- A novel neuro-symbolic framework that extends probabilistic models of pragmatic question answering to more open-ended natural language.
- A systematic investigation of how different integrations of neural and symbolic components affect model behavior.
- Empirical validation against human data, demonstrating that neuro-symbolic models can match or exceed traditional probabilistic approaches in predicting human behavior.

2 A Probabilistic Cognitive Model of Relevant Question-Answering

The probabilistic cognitive model we use for task analysis and scaffolding, which we refer to as the *QA model* (Hawkins et al., to appear), captures a rational *pragmatic respondent* that chooses an answer by reasoning about how a pragmatic questioner chooses a question (see Figure 1 for overview and Appendix A for technical detail). The questioner is grounded in a context-independent *base-level respondent*. The pragmatic questioner selects a question based on the response they expect to get from the base-level respondent, who answers austerely without considering the wider context. The pragmatic respondent, in turn, reasons about the motivation of the speaker for asking the question (i.e., *infers* their goal from the question) and chooses responses that are expected to be relevant to the questioner's goal.

To implement expected relevance of an answer, the QA model builds on decision-theoretic accounts of relevance of questions and answers (van Rooy, 2003; Benz, 2006), which formalizes relevance in terms of a *decision problem (DP)*. The DP includes a real-valued *utility function* of how useful different alternatives (e.g., iced coffee, soda, Chardonnay) are for a given goal (e.g., getting an iced tea). The questioner selects questions that have a high expected relevance (i.e., high *expected utility*) of information from the base-level respondent. The pragmatic respondent uses the questioner's goal-oriented choice of question to infer from the question what kind of DP the questioner likely has. These inferences then guide the respondent's choice of information that will likely increase the expected utility for the questioner, traded off with response costs. We use a probabilistic implementation of the QA model in WebPPL (Goodman and Stuhlmüller, 2014) from Hawkins et al. (to appear) as a starting point and baseline. As commonly done for probabilistic modeling, for these simulations we specified the space of possible answers, possible questions, the literal semantics and the DP utility function specifically for the main experimental materials (see Section 3.1 and Appendix A.1).

Before diving into neuro-symbolic model evaluation, we first validate whether the task decomposition stipulated in the QA model is actually borne out in human intuitive reasoning. To this end, we conducted an exploratory *answer explanation* experiment. Participants (N=50) were recruited via Prolific and shown four trials with contexts wherein a person asked for a target item while several alternative options were available, similar to the initial café example, which constituted the main materials we describe in more detail in Section 3.1. The question was followed by a character replying "no" and providing one, most relevant, competitor alternative. Participants were asked to type an explanation of why that response was reasonable and what would justify mentioning the particular option over a different one. We then analyzed the types of provided explanations, distinguishing between explanations that appealed to (1) abstract similarity of options, (2) questioner goals, desires, intentions, or preferences, and (3) features that were functionally relevant for the questioner goal (e.g., being and iced non-alcoholic drink). If participants spontaneously reason about questioner goals and respective relevant option features as formalized in the QA model, we hypothesize that the proportion of (2) and (3) will be higher than (1). We found that 0.43 of responses appealed to goals (2), 0.20 to goal-relevant features (3), and 0.21 to general similarity (1). 0.13 of responses were unclassifiable (e.g., only appealed to respondent politeness). We interpret this as mild *prima facie* support for the task decomposition implemented in the probabilistic QA model. In the next section, we analyze how systematically replacing different components of the QA model with LLM modules affects the fit to human data.

3 Evaluating Neuro-Symbolic QA models

We investigate the neuro-symbolic framework starting with models where only one component of the task is supplied by an LLM. We then incrementally increase the number of LLM-based modules and change their types, while observing the changes of the *fit to human data* and the *qualitative changes in the predictions*. The driving motivation is to make PCMs more generally applicable (open-ended). For that, two steps are necessary. For one, we would like to be able to generate an in principle open-ended set of alternatives over which to reason or which to choose from. Consequently, we test if LLMs can provide plausible sets of responses, questions, and questioner goals for the QA model; we call LLMs in this role **proposers** (cf. Sumers et al., 2023; Tsvilodub et al., 2024a). For another, once we have open-ended sets of alternatives, we need to be able to obtain information about them for downstream computation, i.e., we also use LLMs in the role of **evaluators** for judging literal semantics of answers and for assessing the utility of options.

3.1 Experimental setup

For all reported simulations below, we use `GPT-4o-mini` for the LLM modules, with the sampling temperature $\tau = 0.1$. All simulations are run for five iterations. We report additional results with the open-source LLM `Qwen-2.5-32B-Instruct` in Appendix D. We use experimental materials, human data and the one-shot LLM prompt from Tsvilodub et al. (2023) to investigate what kinds of alternative options (e.g., iced coffee or Chardonnay), if any, different neuro-symbolic QA models mention in the predicted responses, given a polar question (e.g., "Do you have iced tea?") and different options in context.

The materials include 30 commonsense vignettes similar to the initial barista example. The context always included three possible options, but not the requested target (i.e., iced tea). The options always included a best-fitting alternative called the *competitor* (e.g., iced coffee), a conceptually *similar* option that was deemed less relevant for the questioner's goal (e.g., soda), and an *unrelated* option irrelevant for the uttered request (e.g., Chardonnay). Experimental subjects provided answers by freely typing into a text box. Responses were categorized as "target," "similar," and "unrelated." In addition to these three categories, corresponding to mentioning each of the single options, the categorization also distinguished responses that mentioned *all options*, as well as responses that mentioned *no options*.

If a respondent is engaging in pragmatic reasoning, we would expect her to prefer competitor responses over other types. Tsvilodub et al. (2023) found that humans are, in fact, *relevantly overinformative*, strongly preferring competitor responses (0.52 of responses) over exhaustive responses (0.10), no options responses (0.20), similar (0.18) or unrelated responses (0.00). We investigate how well neuro-symbolic models match human behavior, operationalized via Jensen-Shannon divergence between the observed human data and the models' categorical predictions.

3.2 Integrating LLM Evaluators in the PCM

We assess a class of models that, starting from the QA model, systematically incorporate LLM modules into the PCM architecture which take over two functions: (i) the evaluation of utility of an option, and (ii) the evaluation of the truth of a response. Figure 2 (lower panel) shows a schematic overview

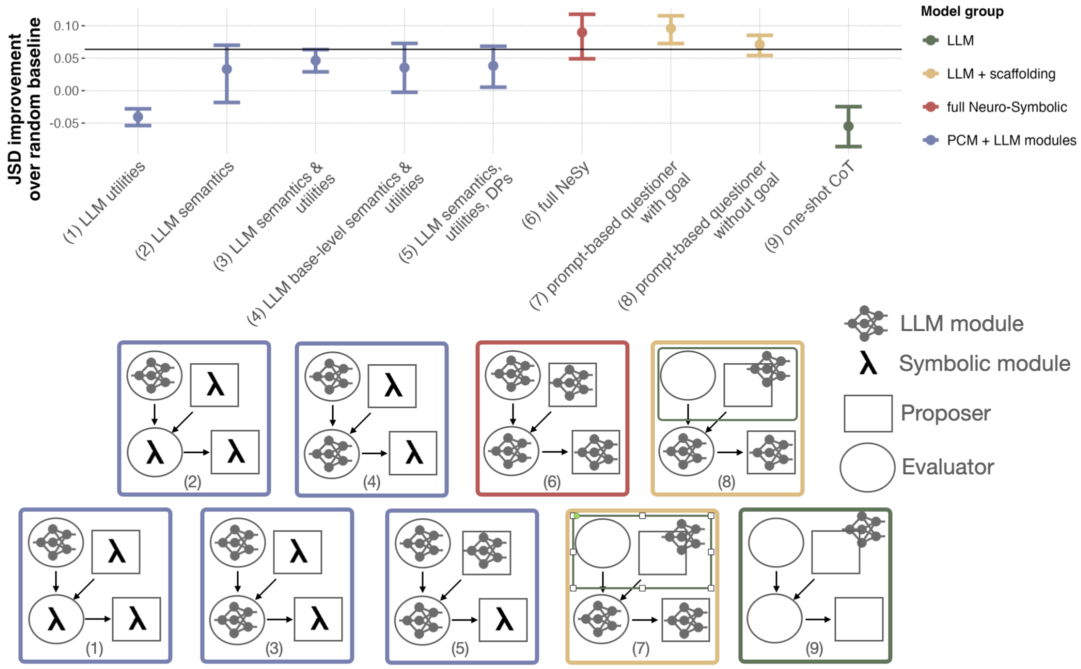


Figure 2: **Upper panel:** Improvement of the model fit to human data in terms of Jensen-Shannon divergence over a uniform response distribution baseline (higher is better, y-axis) of all analyzed models (x-axis). The horizontal line indicates performance of the probabilistic model. Dots indicate the means across simulations, error bars indicate 95% bootstrapped CIs. **Lower panel:** Overview of tested models. Each box shows a schematic of one model, labeled on the x-axis in the plot above it. The models are ordered from closest to the PCM on the left (only one component is LLM-based), to a model only using a single LLM with a single prompt on the right.

of the tested models.

First, we implement an LLM *utility evaluator* for instantiating the utility function in the questioner's decision problem (resulting in the **"LLM utilities" model**). The utility function defines real-valued utilities for the different alternatives (e.g., the iced coffee, soda), conditioned on a target object (e.g., iced tea). In the original QA model, the utilities were elicited in a human rating experiment wherein participants were asked to provide slider ratings for each possible option (e.g., iced tea, iced coffee, soda, Chardonnay), given another option as the goal (see Appendix A.1). To replace the human input with an LLM, we prompted the utility evaluator in a way identical to the instructions of the human elicitation experiment, namely to predict the full space of utilities via ratings on a scale with range 0–100 instead of slider ratings. Importantly, the prompt (and the original human experiment) only asked for abstract ratings, independent of the functional context in which the options occurred in the question answering scenario (see Appendix B for all full prompts). The remaining model components (e.g., the set of alternative utterances, the semantics) remained symbolic in this model.

Beyond replacing the utility component, another function-based component to replace with LLMs for open-ending the PCM is *semantic evaluation*. Semantic evaluation is necessary for the base-level and for the pragmatic respondent and assesses whether a response is true in a particular context. While base-level and pragmatic respondent have slightly different responses at their disposition owing to the fact that the base-level responder is not reasoning about the context (see Appendix A), the semantic evaluation is essentially the same. For an answer like "No, but we have iced coffee." the module has to check whether the polar answer part (e.g., "yes", "no") is true for a context (e.g., the café has soda and iced coffee), given the question (e.g., "Do you have iced tea?"). It also has to evaluate whether the added information (e.g., "We have iced coffee.") is actually correct. We explored models with different combinations of these evaluators. The **"LLM semantics"** model uses an LLM-based semantic evaluator for both the base-level and the pragmatic respondent, while using the same utility component as the original QA model (based on the human experimental data). The **“LLM semantics & utilities”** model employs all described LLM evaluators. The **“LLM base-level semantics & utilities”** only uses an LLM-based base-level respondent, a rule-based pragmatic respondent, and the LLM utility evaluator. The predictions of all models are compared in Section 4.

3.3 Integrating LLM Proposers in the PCM

Next, we integrate LLMs as *proposers* for sets of alternatives required by the QA model. We start with sampling the possible questioner goals with a *goal proposer*. The LLM was prompted to generate plausible text-based goals, given the context and question (see Figure 11). While the set of possible goals in the PCM only contained four DPs (each defining a preference for one of the options: target, competitor, similar, unrelated option), the proposer may sample any text-based questioner goal description. These sampled text-based goals are connected to a DP representation via the *utility evaluator* (Section 3.2). The evaluator was prompted to generate the utilities for the available options, conditioned on each proposed goal. The **“LLM semantics, utilities, DPs”** model uses the goal proposer together with the evaluators from Section 3.2, while the sets of possible utterances and questions are symbolic (i.e., pre-specified manually).

Further open-ending the QA model, we introduce a *response proposer* and a *question proposer* which provide the set of alternative questions and pragmatic answers that the respective pragmatic agents reason over. In both cases, the LLM was concisely prompted to generate n alternatives to an observed utterance or question given the context vignette (see Figure 9, Figure 10). We set $n = 10$ for the response proposer, and $n = 3$ for the question proposer. Here, we address the empirical question whether LLMs, out of the box, can be (easily) prompted to produce the expected types of alternative pragmatic responses in the context of the QA model (no options, competitor, similar, unrelated, all options). Based on exploratory qualitative analyses described in Section 4 in more detail, we append “no-options” and “all-options” responses constructed in a rule-based manner to the set of sampled alternatives. The observed question was always added to the set of sampled alternatives provided by question proposer.

The question and response proposers were tested as part of the fully neuro-symbolic replication of the PCM (**“full NeSy”** model). This model implements the full task decomposition of the QA model, capturing the pragmatic respondent’s recursive reasoning (Figure 1) fully via the modules described above. The base-level respondent uses an LLM-based semantic evaluator to (symbolically) select an informative, true response to a given question (assuming that the decision problem is known). For the pragmatic interpreter, the different possible questions are supplied by an LLM-based question proposer. An LLM-based utility evaluator rates the usefulness of potential options to (symbolically) compute the questioner’s expected utility of each question (based on the expected behavior of the base-level respondent). Finally, the pragmatic respondent estimates likely DPs among the neurally sampled alternatives, given the question, symbolically via Bayes rule (where the likelihood term is approximated via samples of generated questions given a DP). Given her posterior beliefs about the DPs, the respondent chooses a response from the set provided by the response proposer that maximizes her utility function. The respondent’s utility function combines the expected utility of a response with informativeness, formalized as a KL divergence term (see Appendix A for details). We assume flat priors and no utterance costs throughout the model.

3.4 Scaffolding Prompted LLMs with Cognitive Modules

All previous models have implemented computational components suggested by the original QA model with LLM-based proposers and evaluators. These LLM-based components implemented rather “local”, smaller computational elements of the task analysis suggested by the QA model. Alternatively, we may also use LLMs to replace larger chunks of computation, such as the full pragmatic question answering agent, or even the full task analysis captured by the QA model. In the following, we introduce three models that instantiate this general strategy.

We first consider a model called **prompt-based questioner**, of which we consider two versions, one prompted with questioner goals, and one prompted without goals. This model decomposes the pragmatic respondent’s task into its two high-level components suggested by the PCM: inferring the questioner’s goal based on the observed question, and selecting a response that optimizes the questioner’s utility given the inferred DP. We implement a purely prompt-based pragmatic questioner module that supplies the first component. This prompt-based questioner is used by the pragmatic respondent of the "full NeSy" model for inferring the distribution over DPs sampled with an LLM-based goal proposer. The prompt-based questioner takes a questioner goal, the context, and prompts the LLM to provide a likelihood of someone asking the given question (see Fig. 12). The elicited likelihoods for all questions and DPs are then renormalized and used by the pragmatic respondent. We then compare the role of conditioning this module on the goal, and also use a goal-free prompt where the LLM is asked to assess the question likelihood based on the context only (**prompt-based questioner without goal**, see Fig. 13).

For comparison, we also consider a purely *monolithic* prompting of the LLM. In particular, the **one-shot chain-of-thought model** has a chain-of-thought prompt which *verbalizes* the reasoning steps suggested by the QA model in the chain-of-thought for a single example item (see Figure 14). That is, this model is fully LLM-based, using only one call to one neural module (i.e., the LLM).

4 Results

Quantitative results We used the human answer proportions reported in Section 3 as reference and quantitatively compared models in terms of fit to the human data by calculating the Jensen-Shannon divergence (JSD) between the human and the models' predictions. Specifically, we calculated the score Δ_i of model M_i in comparison to the performance of a baseline B given by a flat distribution over all answer categories:

$$\Delta_i = JSD(B, \text{humans}) - JSD(M_i, \text{humans})$$

where $JSD(B, \text{humans}) = 0.154$. We report Δ_i-s in Figure 2 (upper panel; higher JSD differences are better, indicating closer fit to human data). The figure additionally shows the reference value provided by the PCM (solid line).

We found that most tested models with intermediate or high degrees of task decomposition came close to the original PCM (the CIs overlap with the PCM reference line or lie above it), indicating that the neuro-symbolic framework provides a potentially viable method for explaining human data. Visually, the "full NeSy" model and the "prompt-based questioner with goals" fit human data best in terms of Δ. The PCM + LLM models tended to improve with a higher number of LLM modules, but generally provided a somewhat worse fit than the PCM (the means are below the line). Supporting LLMs with a theoretically motivated task decomposition led to significant improvement within the LLM + scaffolding models: the "prompt-based questioner" models showed a better fit than the "one-shot CoT" model. Therefore, overall we found that the neuro-symbolic approach to open-ending pragmatic PCMs showed quantitative fit to human data on par with established cognitive modeling, while offering a more realistic interface to natural language inputs and outputs.

Qualitative results Next to the quantitative analyses, we analyzed qualitatively the differences between model predictions and the performance of the single modules. Figure 3 shows the proportions of different response categories (e.g., competitor, no-options responses etc.) predicted by the different models, next to PCM predictions and human data from Tsvilodub et al. (2023). The figure reveals that although many neuro-symbolic models have similar fit to human data in terms of Δ, there are qualitative differences in the predicted response proportions. The two models with "LLM semantics" overpredicted the proportion of unrelated responses, while the "LLM base-level semantics & utilities" model overpredicted the all-options response rate and slightly underpredicted the competitor rate.

Comparisons of the base-level and pragmatic respondent semantic modules revealed that the base-level semantics module performed reliably, while the pragmatic respondent semantic module made mistakes more frequently, including when evaluating unrelated responses. This may have led to the overprediction of the unrelated responses, as shown by the comparison of the "LLM semantics & utilities" and the "LLM base-level semantics & utilities" models because the former only differs from the latter by using an LLM-based pragmatic respondent semantics evaluator. We correlated the utility evaluator predictions with data elicited from humans for the PCM (see Figure 5) and found a very high correlation ($R = 0.92$), so we can likely rule out the utility evaluator as the source of overprediction of the unrelated category.

The comparison of the PCM + LLM models to the "full NeSy" model highlights the difference in response proportions that is driven by adding LLM proposers for the set of available responses and questions. The addition of response and question proposers decreased the rate of unrelated re-

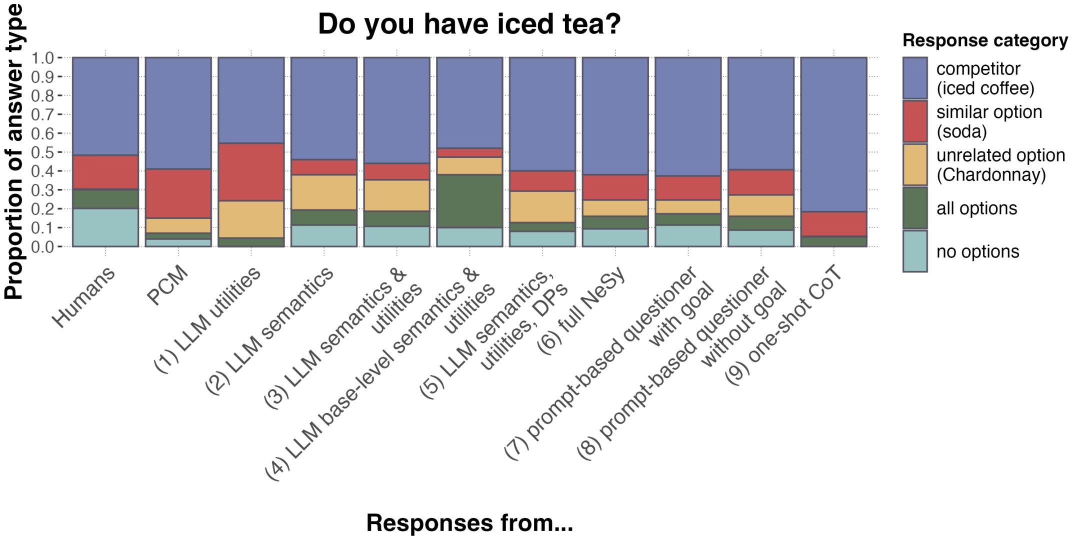


Figure 3: Proportions of different response categories produced by humans (left column) and predicted by different models. The categories are based on which options are mentioned in the response.

sponses and slightly increased the rate of similar and exhaustive responses. Since the "full NeSy" model included the pragmatic respondent semantic evaluator module, we can conclude that semantic evaluations might work more reliably with the LLM's own proposals than with the pre-specified sets of responses and questions. These observations are in line with one of the well-known challenges of neuro-symbolic modeling concerning difficulty of converting between neural and symbolic representations that is required in order to reliably compute truth values for open-ended sentences and contexts (Bader et al., 2004), as well as with debates around LLMs' ability to provide reliable evaluations (Bavaresco et al., 2024).

We also explored decreasing and increasing the n of alternative responses proposed by the LLM. We found that results with $n < 10$ proposals were unlikely to contain the "all options" or "no options" responses. For $n = 10$ this was more often the case, but we appended these two response types to set of alternatives manually nonetheless, to ensure availability of all conceptually meaningful response types. Sampling $n = 50$ responses ensured full coverage of response types but became computationally expensive. Generally the proposals often contained multiple instances of one response type (e.g., multiple competitor responses), an observation we return to in the discussion. However, this is unlikely the sole driving force beyond the fit of the framework, as the "LLM semantics, utilities, DP" model showed a similar competitor response proportion, while operating on a fully prespecified set of responses.

We qualitatively assessed the samples of the goal proposer module that generates possible text-based questioner goals, given the vignette. We compared the samples to human data from a web-based experiment wherein participants were asked to write three plausible goals of the questioner, given the vignette context (see Appendix C for details and human results). We focused on analyzing whether the LLM-proposed goal focused on getting the *target* mentioned in the question, on a more *general* information gain, or on *specific* situation aspects. We observed that, while LLM proposals were plausible, they focused on the target and specific goals around the target more, while humans showed more diversity in their specific goals, e.g., often involving social aspects of the described situation.

Turning to the LLM + scaffolding model type, comparing the "prompt-based questioner model without goals" and the "prompt-based questioner model with goals" revealed a trend towards predicting unrelated and similar responses more uniformly in the goal-free model, which is expected given that the distinction between these types of answers is based on reasoning about the questioner's goal. However, these differences are small and indicate that, even under certain (ablating, from a theoretical perspective) prompt variation, LLMs may be able to approximate pragmatic behavior.

Taken together, our key results are:

- the neuro-symbolic modeling approach fits human data quite closely, potentially making it a framework for computational modeling of pragmatic question answering performing on par with the PCM;

- at least some level of task decomposition when using LLM modules is required for a good fit to human data;

- LLM modules are generally good proposers, although attention should be paid to *types* of proposals that are expected for explanatory purposes;

- LLMs are good evaluators for functions based on abstract world knowledge like the utility evaluator;

- LLMs may struggle with truth-conditional semantics of certain utterances, but perform well when evaluating yes/no responses to polar questions.

5 Related work

Our work is situated at the intersection of several strands of like-minded work in different areas, in addition to the work we build on directly (Hawkins et al., 2015; Tsvilodub et al., 2023). The idea and promise of neuro-symbolic models has been studied in artificial intelligence for many years (Bhuyan et al., 2024). Further, our framework is closely related to recent work outlining various approaches to combining scaffolding structures, computational modeling or cognitive architectures with LLMs (e.g., Nye et al., 2021; Collins et al., 2022; Sumers et al., 2023; Wong et al., 2023; Kambhampati et al., 2024). Combining LLMs with PCMs specifically in the context of computational pragmatics has received some attention in recent work (e.g., Lew et al., 2020; Franke et al., 2024; Tsvilodub et al., 2024a) but the present work focuses specifically on systematically comparing and evaluating families of related models with varying degrees of neural or symbolic computation.

On an algorithmic level, our models combine several LLM calls in a particular architecture, which has been widely used in recent prompt techniques (Nye et al., 2021; Prystawski et al., 2023; Yao et al., 2023), and systems that use LLM calls to retrieve information (e.g., Lewis et al., 2020), to access different tools (e.g., Schick et al., 2023) or to solve complex reasoning tasks (e.g., Creswell et al., 2022; He-Yueya et al., 2023).

Systems with multiple LLM calls per input have also been specifically applied to question answering (Wang et al., 2023), mainly with a focus on improving factual accuracy of responses, or on training systems to improve their question asking capabilities (Andukuri et al., 2024). Therefore, our case study addresses a highly relevant task, with a novel focus on modeling *pragmatic, human-like* answering behavior.

6 Discussion

Taken together, in this case study we outlined and systematically assessed a neuro-symbolic framework for computational pragmatic modeling that uses probabilistic cognitive models as scaffolding structure that integrates LLM components for more flexible interfaces with language and background knowledge. The experiments on a case study of pragmatic question answering revealed that such modeling can be a viable candidate in the toolbox for more flexible models of human behavior in question answering. The systematic comparison of neuro-symbolic models with different degrees of task decomposition suggests fine-grained differences in how LLMs perform on different subtasks common to PCMs.

Our case study has several limitations, but also opens up paths for future work. For one, the full neuro-symbolic models implement Bayesian inference via enumeration, which results in computational bottlenecks when scaling the number of proposals and options in context. Related work connecting LLMs and Bayesian inference might be a promising avenue for improvements (Lew et al., 2023). Additionally, the current main results are based only on one closed-source LLM (but see Appendix D for exploratory results with an open-source LLM), and only use zero-shot prompting (except the CoT model). In this initial case study, we prioritized using relatively simple, non-engineered prompts, but nonetheless LLM prompting comes with potential risks of hallucination, errors and biases (e.g., Bender et al., 2021; Ji et al., 2023; Liu et al., 2023).

Finally, the use of LLMs as proposers and evaluators opens up interesting questions. For instance, response proposals supplied by the LLM might contain a trend towards certain response types, which can arguably be seen as a learned prior over human preferences reflected in the training data. Additionally, cognitive models usually assume utterance costs for human language production and comprehension, but such online processing costs might not have a clear counterpart in LLMs. Further, varying performance of LLM evaluators might suggest that some aspects of semantics might be amortized in training data (White et al., 2020). Our results suggest that LLMs might not approximate different aspects of human intuitive knowledge equally well, touching upon important considerations of replacing human judgements with LLMs (Shiffrin and Mitchell, 2023; Löhn et al., 2024). For the LLMs + PCM models, one other potential source of improved performance with scaffolding of the LLM could be due to higher inference time compute budget that comes with decomposing the task into several LLM calls (Yu et al., 2024).

In sum, we presented a detailed case study as a starting point for exploring neuro-symbolic models of human language use, showing that task decomposition supplied by a cognitive model can be leveraged in synergy with recent LLMs, working towards open-ending pragmatic computational modeling.

Acknowledgments

We would like to thank Fausto Carcassi for his contributions to developing the framework, and the anonymous reviewers for insightful comments. MF is a member of the Machine Learning Cluster of Excellence at University of Tübingen, EXC number 2064/1 – Project number 39072764. PT and MF gratefully acknowledge the support by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant INST 35/1597-1 FUGG.

References

Chinmaya Andukuri, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah Goodman. 2024. STar-GATE: Teaching language models to ask clarifying questions. In *First Conference on Language Modeling*.

Sebastian Bader, Pascal Hitzler, and Steffen Hoelldobler. 2004. The integration of connectionism and first-order knowledge representation and reasoning as a challenge for artificial intelligence. *Preprint*, arXiv:cs/0408069.

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, et al. 2024. LLMs instead of human judges? A large scale empirical study across 20 NLP evaluation tasks. *URL https://arxiv.org/abs/2406.18403*.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Margaret Mitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Anton Benz. 2006. *Utility and relevance of answers*. Springer.

Bikram Pratim Bhuyan, Amar Ramdane-Cherif, Ravi Tomar, and TP Singh. 2024. Neuro-symbolic artificial intelligence: a survey. *Neural Computing and Applications*, pages 1–36.

Herbert H Clark. 1979. Responding to indirect speech acts. *Cognitive psychology*, 11(4):430–477.

Katherine M. Collins, Catherine Wong, Jiahai Feng, Megan Wei, and Joshua B. Tenenbaum. 2022. Structured, flexible, and robust: benchmarking and improving large language models towards more human-like behavior in out-of-distribution reasoning tasks. *Preprint*, arXiv:2205.05718.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*.

Judith Degen. 2023. The rational speech act framework. *Annual Review of Linguistics*, 9(1):519–540.

Simon Farrell and Stephan Lewandowsky. 2018. *Computational modeling of cognition and behavior*. Cambridge University Press.

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

Michael Franke, Polina Tsvilodub, and Fausto Carcassi. 2024. Bayesian statistical modeling with predictors from LLMs. *arXiv preprint arXiv:2406.09012*.

Noah D Goodman and Andreas Stuhlmüller. 2014. The Design and Implementation of Probabilistic Programming Languages. `http://dippl.org`. Accessed: 2025-1-30.

Jeroen Antonius Gerardus Groenendijk and Martin Johan Bastiaan Stokhof. 1984. *Studies on the Semantics of Questions and the Pragmatics of Answers*. Ph.D. thesis, Univ. Amsterdam.

Auli Hakulinen. 2001. Minimal and non-minimal answers to yes-no questions. *Pragmatics*, 11(1):1–15.

CL Hamblin. 1973. Questions in Montague English. *Foundations of Language*, 10(1):41–53.

Robert D. Hawkins and Noah D. Goodman. 2017. Why do you ask? The informational dynamics of questions and answers. PsyArXiv.

Robert D. Hawkins, Andreas Stuhlmüller, Judith Degen, and Noah D. Goodman. 2015. Why do you ask? Good questions provoke informative answers. *Cognitive Science*.

Robert D. Hawkins, Polina Tsvilodub, Claire Augusta Bergey, Noah D. Goodman, and Michael Franke. to appear. Relevant answers to polar questions. *Philosophical Transactions B*.

Joy He-Yueya, Gabriel Poesia, Rose E Wang, and Noah D. Goodman. 2023. Solving math word problems by combining language models with symbolic solvers. *arXiv preprint arXiv:2304.09102*.

Jennifer Hu, Sammy Floyd, Olessia Jouravlev, Evelina Fedorenko, and Edward Gibson. 2023. A fine-grained comparison of pragmatic language understanding in humans and language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4194–4213, Toronto, Canada. Association for Computational Linguistics.

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

Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Saldyt, and Anil Murthy. 2024. LLMs can't plan, but can help planning in LLM-Modulo frameworks. *Preprint*, arXiv:2402.01817.

Alexander K Lew, Michael Henry Tessler, Vikash K Mansinghka, and Joshua B Tenenbaum. 2020. Leveraging unstructured statistical knowledge in a probabilistic language of thought. In *Proceedings of the annual conference of the cognitive science society*.

Alexander K. Lew, Tan Zhi-Xuan, Gabriel Grand, and Vikash K. Mansinghka. 2023. Sequential Monte Carlo steering of large language models using probabilistic programs. *Preprint*, arXiv:2306.03081.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Preprint*, arXiv:2307.03172.

Lea Löhn, Niklas Kiehne, Alexander Ljapunov, and Wolf-Tilo Balke. 2024. Is machine psychology here? On requirements for using human psychological tests on large language models. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 230–242, Tokyo, Japan. Association for Computational Linguistics.

Maxwell Nye, Michael Tessler, Josh Tenenbaum, and Brenden M Lake. 2021. Improving coherence and consistency in neural sequence models with dual-system, neuro-symbolic reasoning. *Advances in Neural Information Processing Systems*, 34:25192–25204.

Kathryn Pruitt and Floris Roelofsen. 2011. Disjuntive questions: Prosody, syntax, and semantics. Handout, Göttingen.

Ben Prystawski, Paul Thibodeau, Christopher Potts, and Noah Goodman. 2023. Psychologically-informed chain-of-thought prompts for metaphor understanding in large language models. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 45.

Anselm Rothe, Brenden M Lake, and Todd Gureckis. 2017. Question asking as program generation. *Advances in neural information processing systems*, 30.

Laura Ruis, Akbir Khan, Stella Biderman, Sara Hooker, Tim Rocktäschel, and Edward Grefenstette. 2023. The goldilocks of pragmatic understanding: Fine-tuning strategy matters for implicature resolution by LLMs.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Preprint*, arXiv:2302.04761.

Richard Shiffrin and Melanie Mitchell. 2023. Probing the psychology of AI models. *Proceedings of the National Academy of Sciences*, 120(10):e2300963120.

Jon Scott Stevens, Anton Benz, Sebastian Reuße, and Ralf Klabunde. 2016. Pragmatic question answering: A game-theoretic approach. *Data & Knowledge Engineering*, 106:52–69.

Theodore R Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L Griffiths. 2023. Cognitive architectures for language agents. *arXiv preprint arXiv:2309.02427*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Polina Tsvilodub, Michael Franke, and Fausto Carcassi. 2024a. Cognitive modeling with scaffolded LLMs: A case study of referential expression generation. In *ICML 2024 Workshop on LLMs and Cognition*.

Polina Tsvilodub, Michael Franke, Robert Hawkins, and Noah D. Goodman. 2023. Overinformative question answering by humans and machines. In *Proceedings of the 45th Annual Conference of the Cognitive Science Society*. Cognitive Science Society.

Polina Tsvilodub, Paul Marty, Sonia Ramotowska, Jacopo Romoli, and Michael Franke. 2024b. Experimental pragmatics with machines: Testing LLM predictions for the inferences of plain and embedded disjunctions. In *Proceedings of CogSci*, pages 3960–3967.

Robert van Rooy. 2003. Questioning to resolve decision problems. *Linguistics and Philosophy*, 26(6):727–763.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Julia White, Jesse Mu, and Noah D. Goodman. 2020. Learning to refer informatively by amortizing pragmatic reasoning. *Preprint*, arXiv:2006.00418.

Li Siang Wong, Gabriel Grand, Alexander K. Lew, Noah D. Goodman, Vikash K. Mansinghka, Jacob Andreas, and Joshua B. Tenenbaum. 2023. From word models to world models: Translating from natural language to the probabilistic language of thought. *ArXiv*, abs/2306.12672.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Preprint*, arXiv:2305.10601.

Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. 2024. Distilling system 2 into system 1. *Preprint*, arXiv:2407.06023.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2023. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*.

A QA model

Below, we report the QA model by Hawkins et al. (to appear), described in Section 2, in more formal detail.

The base-level respondent that provides literal responses r to a question q given the world w is defined as follows:

$$R_0(r \mid w, q) \propto \begin{cases} 1 & \text{if } r \text{ is true in } w \text{ \& safe for } q \\ 0 & \text{otherwise.} \end{cases}$$

The notion of safety is couched in prior work on semantics of questions and answers (Pruitt and Roelofsen, 2011) and entails that, for the tested vignettes, only the literal answers $r \in \{\text{'yes'}, \text{'no'}\}$ are evaluated here.

The pragmatic questioner selects a question given their decision problem, based on the responses they expect from the base-level respondent R_0. Formally, a *decision problem* (DP) is a tuple $\mathcal{D} = \left\langle \mathcal{W}, \mathcal{A}, \mathcal{U}, \pi_Q^{\mathcal{W}} \right\rangle$, consisting of a set of world states $\mathcal{W}$, a set of options $\mathcal{A}$, a utility function $\mathcal{U} : \mathcal{W} \times \mathcal{A} \to \mathbb{R}$, and a probability distribution $\pi_Q^{\mathcal{W}} \in \Delta(\mathcal{W})$ capturing the questioner's prior beliefs about the world states. Then, the *value of a decision problem* $\mathcal{D}$ is the expected utility under a policy $\aleph^{\mathcal{D}}$ that chooses options according to their expected utility:

$$V(\mathcal{D}) = \mathop{\mathbb{E}}_{a \sim \aleph^{\mathcal{D}}} \left[\mathop{\mathbb{E}}_{w \sim \pi_Q^{\mathcal{W}}} \Big[\mathcal{U}(w, a) \Big] \right]$$

The pragmatic questioner then selects a question by soft-maximizing the expectation over the values of the decision problems $\mathcal{D}^{|r,q}$ given likely responses from the base-level respondent, resulting in $Q(q \mid D)$ (see Figure 4), where C(r) and C(q) are the production costs associated with the response and question, respectively.

The pragmatic respondent then reasons about the pragmatic questioner's choice of question in order to infer their likely decision problem:

$$\pi_{R_1}^{\mathfrak{D}|q}(\mathcal{D}) \propto Q(q \mid \mathcal{D}) \, \pi_{R_1}^{\mathfrak{D}}(\mathcal{D})$$

Finally, the pragmatic respondent chooses a response by soft-maximizing the expected utility of the response given their posterior beliefs about the questioner DP. Utility is defined as a (parameterized) combination of informativity (defined via KL divergence) and action-relevance (defined via the decision problem value), resulting in $R_1(r \mid q)$ (see Figure 4).

A.1 Parameterization of the QA model

As commonly done for probabilistic modeling, in order to run simulations with the QA model parameters of the model were specified by the modelers or with elicited human data (Hawkins et al., to appear). For each vignette, the set of alternative questions included polar questions about the availability of each of the possible options individually, and a wh-question inquiring about all possible options.

$$Q(q \mid \mathcal{D}) = \underset{\alpha_Q}{\mathrm{SM}} \left(\underset{w \sim \pi_Q^{\mathcal{W}}}{\mathbb{E}} \left[\underset{r \sim R_0(\cdot \mid w, q)}{\mathbb{E}} \left[V(\mathcal{D}^{\mid r,q}) - \mathrm{C}(r) \right] \right] - \mathrm{C}(q) \right)$$

$$R_1(r \mid q) = \underset{\alpha_R}{\mathrm{SM}} \left(\underset{\mathcal{D} \sim \pi_{R_1}^{\mathcal{D} \mid q}}{\mathbb{E}} \left[(1 - \beta)\, (-\mathrm{KL} \left(\pi_Q^{\mathcal{W}^{\mid r,q}} \mid\mid \pi_{R_1}^{\mathcal{W}} \right)) \;\; + \beta\, V \left(\mathcal{D}^{\mid r,q} \right) - \mathrm{C}(r)) \right] \right)$$

Figure 4: Formal definitions of the pragmatic questioner $Q(q \mid D)$ and respondent $R_1(r \mid q)$.

The set of available pragmatic answers included answers of all categories described in Section 3.1.

In order to specify the utility functions of the questioner DPs, a web-based experiment was run with human participants. Participants ($N = 453$) were asked to provide slider ratings for each possible option (e.g., iced tea, iced coffee, soda, Chardonnay), given another option as the goal. The full space of possible combinations was elicited. The slider ratings were on a scale of 0–100. Importantly, participants were asked to rate how happy they think a person would be to receive an option, given the target, resulting in *abstract* conditional preferences. The DP utilities for each vignette were bootstrapped from human preferences in the QA model simulations. Human results for ratings of the alternatives, given the option used as the target in the free production experiments as the goal (e.g., the iced tea) are shown in Figure 5 (left) together with respective LLM module predictions. Human and `GPT-4o-mini` ratings correlated highly, and supported the intuitive ordering of the relevance of alternatives (e.g., the competitor received higher ratings than the unrelated option for a given target).

B Prompts

Prompts for all LLM modules are presented below in Figures 6–14.

B.1 Semantic Evaluators

The base-level semantic evaluator only evaluates the set of literal responses {'yes', 'no'}. The pragmatic respondent semantic evaluator evaluates the set of possible overinformative responses. In models where the set of pragmatic responses is prespecified, the possible responses are of the form "I'm sorry, we don't have {target}. {continuation}", where the continuation was constructed for all response types (no-options, competitor, similar, unrelated, all-options responses).

C Human Experiment on Goal Inference

In an exploratory *goal inference* study, participants (N=35) were shown vignette contexts without the available options, followed by the question asked by a speaker. Participants were asked to name three plausible goals in three separate text fields that the questioner might have in mind when asking the question. We focused on distinguishing whether participants named goals focused on acquiring the *target* mentioned in the question, on acquiring more *general* information, or on goals related to more *specific* aspects of the situation.

Participants were most likely to infer *specific* goals (0.42 of the responses), followed by *target-related* goals (0.35 of the responses). More *general* information-seeking goals were less likely (0.17 of the responses), and some responses were non-classifiable (0.06).

We then manually analyzed the proposals of the LLM goal proposer module. Qualitatively, the target-related goals mostly were about acquiring the target or an item with the same functional features (e.g., when the target was veggie pizza, the functional feature would be being a vegetarian option), both for humans and LLMs. The specific goals produced by humans often involved more details than just acquiring the target, e.g., acquiring the target for a friend, or mentioned different specific preferences participants came up with. In contrast, the specific goals produced by LLMs were less likely to mention social aspects like acquiring something for a friend, and more likely to produce possible more specific questioner preferences (e.g., "asking about certain dietary restrictions"). The more general goals produced by humans and LLMs often mentioned learning about the set of available alternatives.

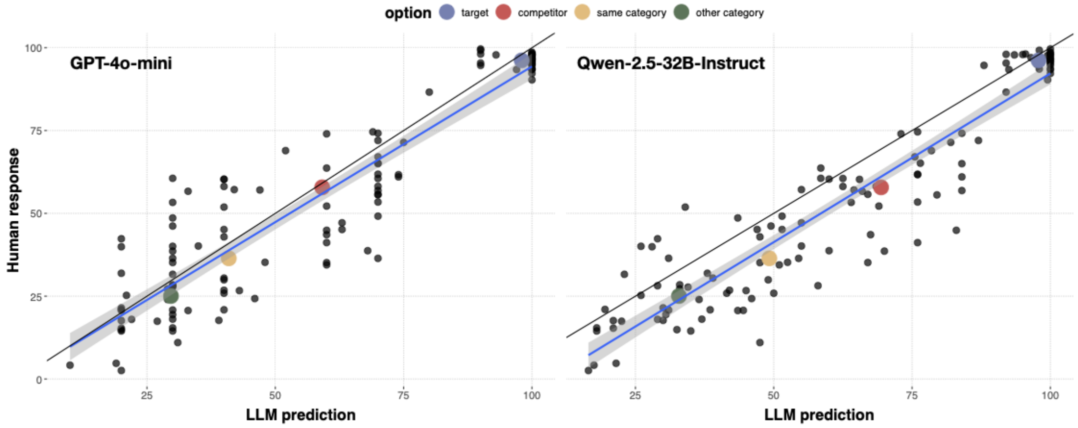


Figure 5: **Left**: GPT-4o-mini utilities plotted against human utilities, $R = 0.92$. **Right**: Qwen-2.5-32B-Instruct utilities plotted against human utilities, $R = 0.93$.

	1-shot CoT	1-shot example	1-shot explanation	0-shot
Qwen-2.5-32B-Instruct	0.21	0.15	0.25	0.28
Qwen-2.5-14B-Instruct	0.16	0.24	0.22	0.39
Qwen-2.5-7B-Instruct	0.33	0.19	0.50	0.17

Table 1: Jensen-Shannon divergence between human response proportions and the proportions of different response categories predicted by Qwen models of different sizes under various prompting (lower is better).

Utility Evaluator Prompt

```
In this study we are interested in how you think about other people.
On each trial, you will be given some information about a person: 'Suppose someone wants to have Italian food.'

Then we'll ask how happy you think this person would be about other things, given this information. For instance, we might ask: 'How happy do you think they would be if they had French food instead?'
You'll use ratings from 0-100 to answer the questions. Return the rating only.

Suppose someone wants {goal}. How happy do you think they would be if they got {option}?
```

Figure 6: **Utility Evaluator Prompt**

Base-level Evaluator Prompt

```
Safe answers to questions only provide information that the questioner genuinely does not know, given what they asked.
True answers to questions only provide information that is true given the context.

Here is an everyday situation where someone asks a question: {context + question}
Here is a potential answer to the question: {utterance}

Is the answer safe and true in this context, according to the definition above?
Return 'yes' or 'no' only.
```

Figure 7: **Base-level Evaluator Prompt**

Pragmatic Respondent Semantic Evaluator Prompt

```
True answers to questions only provide information that is true given the context.

Here is an everyday situation where someone asks a question: {state}
Here is a potential answer to the question: {utterance}

Is the answer true in this context, according to the definition above?
Return 'yes' or 'no' only.
```

Figure 8: **Pragmatic Respondent Semantic Evaluator Prompt**

Response Proposer Prompt

```
Safe answers to questions only provide information that the questioner genuinely does not know, given what they asked.
True answers to questions only provide information that is true given the context.

Here is a question someone could ask in an every day situation: {question}
Here are the available options: {options}

Generate {num_samples} literal answers to the question.
Return them as a numbered list.
```

Figure 9: **Response Proposer Prompt**

Question Proposer Prompt

```
Suppose a person has the following goal: {goal}
The person is in the following everyday situation: {context}
Generate {num_samples} well formed short questions(s) the person might naturally ask in the context to achieve their goal.
```

Figure 10: **Question Proposer Prompt**

Goal Proposer Prompt

```
You will be given a context in which a person asks a question.
What plausible different goals might the person be interested in, given what they asked?
Your task is to generate {num_samples} alternatives in a comma separated list.
```

Figure 11: **Goal Proposer Prompt**

Prompt-based questioner with goals

```
We are interested in how likely a person would be to ask the following question in a simple context, given their goal.
Please return only the likelihood, provided on a scale between 0 and 1.
Goal: {goal}
Context: {state}
{utterance}
```

Figure 12: **Prompt-based questioner with goals**

Prompt-based questioner without goals

```
We are interested in how likely a person would be to ask the following question in a simple context.
Please return only the likelihood, provided on a scale between 0 and 1.
Context: {state}
{utterance}
```

Figure 13: **Prompt-based questioner without goals**

One-shot chain-of-thought prompt

```
You are hosting a barbecue party. You are standing behind the barbecue. You have the following goods to offer: pork sausages, vegan burgers, grilled potatoes and beef burgers.
Someone asks: Do you have grilled zucchini?

Let's think step by step. You reason about what that person most likely wanted to have. That they asked for grilled zucchini suggests that they might want vegetarian food. From the items you have pork sausages and beef burgers are least likely to satisfy the persons desires. Vegan burgers and grilled potatoes come much closer. Grilled potatoes are most similar to grilled zucchini. You reply:

I'm sorry, I don't have any grilled zucchini. But I do have some grilled potatoes.
```

Figure 14: **One-shot chain-of-thought prompt**

D Simulation Results with an Open-Source LLM

Additionally to the main experiments performed with `GPT-4o-mini`, we ran all experiments with an open-source LLM — `Qwen-2.5-32B-Instruct` (Team, 2024), providing insights about advantages and open questions for our neuro-symbolic modeling framework when it is based on LLMs that can be run locally.

The experimental settings were the same as reported in 3.1. Quantitative results comparing the predictions of the different models to human results in terms of JSD improvement over a random baseline Δ, introduced in 4, are shown in Figure 15. The results indicate that some models with LLM evaluators (i.e., semantics and utility evaluators, models (1) and (3)) perform on par with the models based on a powerful closed-source LLM, as well as close to the original probabilistic model. The high correlation between DP utilities predicted by Qwen and human results (Figure 5, right) corroborates that such evaluations can also be reliably elicited from an open-source model. Similarly to GPT-based models, the performance of the utility evaluator was more robust than for the literal semantic evaluators, as indicated by the better fit to human data for model (1). However, for model (2) and for models introducing a proposer (models (4)–(5)) the fit of the models decreased. Manual evaluations of the single modules in these models indicated that, qualitatively, the generated evaluations and proposals were adequate for the respective modules. However, this LLM struggled more to follow formatting instructions, so that processing the proposals for passing them to the neural evaluator modules was more brittle. Simulation runs which resulted in unrecoverable parsing errors were excluded form analysis.[2] Models which use a Qwen-based prompted questioner module ((6)–(7)) improved the fit to human data over the random baseline, although the role of conditioning the questioner prompt on the goal was opposite to the GPT-based models.

Qualitative results comparing the proportions of different response types under different models are shown in Figure 16. The qualitative patterns suggest that Qwen-based models preferred responses mentioning a relevant alternative (i.e., competitor responses) over no options or exhaustive responses.

[2] For this reason, no results of the full neuro-symbolic model are reported.

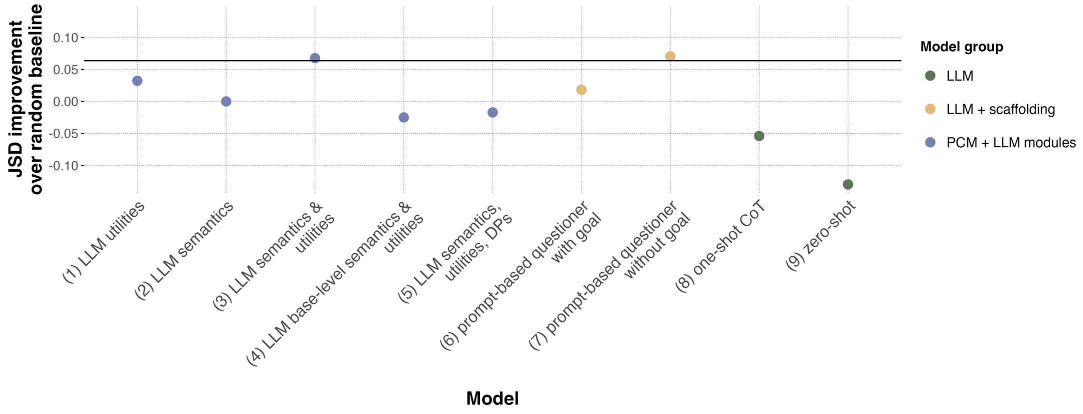


Figure 15: Improvement of the fit to human data of a model with an open-source Qwen-2.5-32B-Instruct backbone over a uniform response distribution baseline (higher is better). The horizontal line indicates the performance of the symbolic probabilistic model. The points indicate averages over simulations.

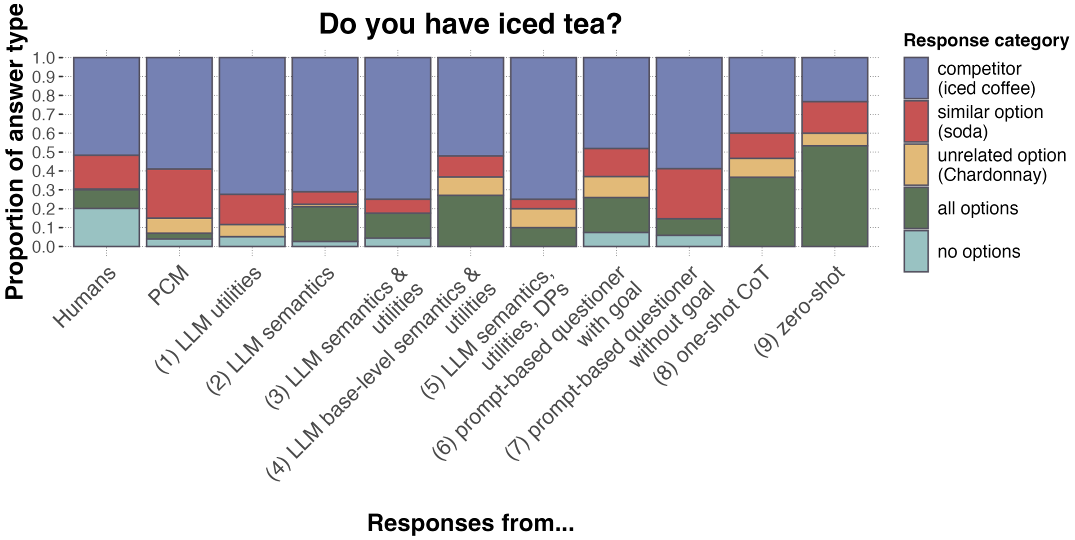


Figure 16: Proportions of different response categories predicted by Qwen-2.5-32B-Instruct used in different models (1–7), and with different prompting strategies (8–9).

LLM-only predictions, both in the one-shot chain-of-thought and the zero-shot prompting conditions, on the other hand, showed a larger proportion of exhaustive responses. We also report the JSD values for predictions from different sizes of Qwen under different prompting strategies from Tsvilodub et al. (2023) and human results in Table 1. These results suggest variation in the effectiveness of such prompting for different model sizes. For the two larger models, prompts that verbalize the PCM improve results over zero-shot prompting, although for the 32B model, ablated prompts further improve the fit to human data, suggesting substantial variation of human-likeness of the predictions when using only neural modules.

In sum, most neuro-symbolic Qwen-based models scaffolded with the PCM showed a better fit to human data than the random baseline, while the predictions of the LLM alone, even under one-shot chain-of-thought prompting, showed worse fit than the baseline. Additionally, given the open availability of the LLM, light-weight fine-tuning for better formatting instruction-following might offer a promising avenue for more robust neuro-symbolic modeling with open-source LLMs. Therefore, we can cautiously conclude that, given sufficient instruction-following capabilities for formatting, the neuro-symbolic framework might allow open-source LLMs to produce more human-like response patterns.

Annotator disagreement in RST annotation schemes

Daniil Ignatev[1], Denis Paperno[1], Massimo Poesio[1,2],

[1]Utrecht University, [2]Queen Mary University of London,

Correspondence: d.ignatev@uu.nl

Abstract

Discourse parsing within the Rhetorical Structure Theory (RST) framework has inspired extensive research; however, it remains prone to significant levels of annotator disagreement, particularly in the labeling of relations and nuclearity. This paper investigates systematic discrepancies in RST annotations, focusing on two expert-annotated corpora of closely related languages. We first compare different RST treebanks to assess the availability of parallel-labeled data and highlight their usefulness for studying disagreement. We then perform both quantitative and qualitative analyses of annotation divergences, identifying factors that contribute significantly to inconsistent interpretations. Finally, we propose two practical approaches for addressing disagreement: (1) filtering out unhelpful biases and (2) capturing legitimate ambiguity through more flexible annotation schemes.

1 Introduction

In the field of computational linguistics, discourse parsing — particularly within the Rhetorical Structure Theory (RST) framework — offers a well-established approach to analyzing the coherence relations between different parts of a text. This task involves identifying and classifying discourse relations, such as the cause-effect relationship, between individual units, like sentences or paragraphs. Foundational work by Mann and Thompson (Mann and Thompson, 1988) and advancements by Daniel Marcu (Marcu, 1996, 2000) have introduced methodologies for constructing trees that represent discourse units and their connections, ultimately reflecting the rhetorical composition of texts. In RST, elementary discourse units (DUs) are roughly analogous to clauses, but higher order units can span indefinitely up to a complete text. The framework employs 30 relations to capture the full range of connections between these units. Related spans are classified into nucleus and satellite, where the nucleus represents the central or more significant unit of the relation[1].

The complexity inherent in discourse annotation frequently leads to disagreements among annotators at multiple levels. Even rigorously designed RST corpora, such as RST-DT (Lynn Carlson, 2002), the Potsdam Commentary Corpus (Stede and Neumann, 2014), and the Dutch Discourse Treebank (van der Vliet et al., 2011; Redeker et al., 2012), typically yield kappa scores reflecting at best substantial agreement.

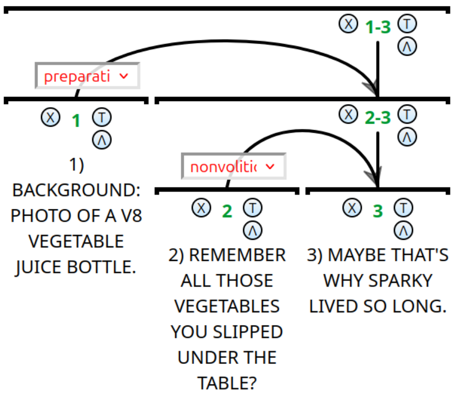

1-3
preparati
1
2-3
nonvolitic
2
3
1) BACKGROUND: PHOTO OF A V8 VEGETABLE JUICE BOTTLE.
2) REMEMBER ALL THOSE VEGETABLES YOU SLIPPED UNDER THE TABLE?
3) MAYBE THAT'S WHY SPARKY LIVED SO LONG.

Figure 1: Example from RST website (Taboada and Mann, 2006) in RSTWeb (Zeldes, 2016). Cropped labels: preparation, nonvolitional cause

On the other hand, while the subject of disagreement in discourse annotation has been widely addressed in theory, there have been relatively few suggestions on how this issue could be addressed in practice. Meanwhile, recent years have seen

[1] Beyond RST, other frameworks such as the Penn Discourse Treebank (PDTB, Prasad et al. 2008) and Segmented Discourse Representation Theory (SDRT, Asher and Lascarides 2003) have explored alternative approaches to labeling discourse relations. For the former, there exists a body of work dealing with disagreement (Yung et al., 2024; Scholman and Demberg, 2017), showing that this problem is relevant for either framework.

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 18-32.
Eugene, Oregon, July 18-20, 2025

the emergence of a large body of work on learning from disagreement, proposing a number of approaches to handling varying interpretations. In natural language processing, transformer-based architectures (Devlin et al., 2019; Liu et al., 2019) are increasingly used to capture nuanced linguistic phenomena, and this includes work on leveraging label distributions and annotator-specific biases (Rodrigues and Pereira, 2017; Mostafazadeh Davani et al., 2022). Such strategies include augmenting the gold standard based on the spectrum of opinions (Plank et al., 2014; Fornaciari et al., 2021), learning from distributions of labels using a soft metric (Sheng et al., 2008; Aroyo and Welty, 2014; Peterson et al., 2019; Uma et al., 2020), and training separate models on labels coming from individual annotators (Akhtar et al., 2020). However, despite these trends, deeper engagement with disagreements in discourse-level tasks like RST parsing has been limited.

Given this tendency, addressing the research gap mentioned above becomes increasingly important. To this end, we pursue several objectives in this paper:

- Review existing RST resources with respect to the extent of disagreeing annotations they contain.
- Perform quantitative and qualitative analyses of factors contributing to disagreement, using suitable data sources.
- Based on the obtained results, propose preferable ways of integrating disagreements into RST annotation and RST parsing.

The scope of this paper primarily concerns RST relations and nuclearity, leaving aside two other major aspects of RST: segmentation of text into EDUs and organizing these segments into spans. While these areas are also subject to disagreement and require thorough analysis, we exclude them here for several reasons. Firstly, in most existing corpora, inter-annotator agreement on these tasks is much higher compared to relation and nuclearity labeling (see Das et al. 2017 for details). Additionally, in most flavors of RST annotation, EDU segmentation is grounded in syntax and leaves considerably less room for subjective interpretation. This is evident to the extent that some RST parsers assume text segmentation is given; while debatable, this assumption remains widely adopted in practical applications (Maekawa et al., 2024).

Our results suggest that RST annotation is substantially influenced by individual preferences of annotators, which sometimes conflict with the annotation manual. In such cases, considering the entire range of disagreeing annotations seems redundant. On the other hand, a larger portion of disagreements is prompted by factors that allow for multiple interpretations, making the adoption of a spectrum of readings by individual experts a generally feasible strategy.

2 Related Work

2.1 Theories of disagreements in discourse annotation

The subject of discrepancies in RST analysis has been widely discussed in the community, with particular attention given to the relational level.

In this context, two notions need to be distinguished: first, one annotator assigning multiple complementary relations; second, several annotators assigning multiple relations that may or may not be complementary. We will refer to the former as "multi-level" annotation and the latter as "disagreement." While our primary focus is on the latter, the concept of multi-level analysis suggests that diverging concurrent analyses may all be plausible: if one annotator can assign multiple complementary relations to the same span, it is reasonable to assume that several annotators can do the same. For this reason, we consider the respective arguments in the discussion, even though they do not concern disagreement directly.

(1) The topic of multi-level analysis has been widely discussed in the literature. For example, Mann and Thompson, 1988 suggested that multiple relations can be assigned to the same span. Similarly, Moore and Pollack, 1992 argued that each relation between rhetorical units should be annotated on two levels: informational and intentional, as the existing relation types exhibit significant overlap with respect to these domains. Arguments in favor of multi-level annotation have since appeared in numerous works (see Taboada and Mann, 2006 for a systematic overview).

However, Sanders and Spooren, 1999, followed by Stede, 2008a, oppose this suggestion, claiming that complex annotation would be redundant in most cases, as the relations involved are typically either exclusively informational or exclusively intentional. Meanwhile, Taboada and Mann, 2006 notes that postulating multiple relations may be justified in ambiguous cases that cannot be resolved based on the context.

(2) The issue of ambiguity in the RST framework has been directly addressed in several works by Manfred Stede (Dipper and Stede, 2006; Stede, 2008b,a), based on the experience of building the Potsdam Commentary Corpus (Stede and Neumann, 2014). The results of this work are summarized in Stede, 2008a, which identifies several sources of ambiguity in RST annotations, such as vagueness in definitions and conflicting scopes of relations, and argues that many of these can be resolved through distinguishing several levels of discourse annotation: thematic, referential, and others. To that end, the work introduces a specialized framework, MLA.

A related line of work (Iruskieta et al., 2015; Wan et al., 2019) proposed changes to how the similarity of structures should be measured in RST annotations. The alternative metrics penalize discrepancies on different levels (relation directionality, nuclearity, relation type) differently, depending on how important each factor is for the overall structure.

Finally, some recent works suggest a permissive approach to concurrent interpretations, advocating for their incorporation into the gold standard. (Das et al., 2017) compare amateur and expert RST annotations in English and German and propose treating competing expert analyses as a “complex ground truth.” They suggest Underspecified Rhetorical Markup Language (URML, Reitter and Stede, 2003) as a means of storing discourse graphs. On the other hand, eRST, a proposal for RST enhancement, allows for additional edges, i.e., concurrent relations, in RST structures, provided these relations are realized lexically through discourse markers. Although this notion does not directly address disagreements, it enables the integration of several alternative analyses into one structure and permits at least some alternative readings on the relational level. In other words, parallel annotations in existing corpora can partially be integrated into eRST graphs.

2.2 Analyzing Annotation Discrepancies

Qualitative analyses of disagreements have primarily been conducted by corpus designers. For instance, da Cunha et al., 2011 examined disagreements in Spanish RST. A significant amount of qualitative analysis of RST disagreements, which ultimately remained unpublished, was carried out by the authors of the Dutch Discourse Treebank (NLDT) based on their own material. While we conducted our qualitative analysis independently on a subset of their corpus, resulting in different hypotheses, we extend our gratitude to Gisela Redeker for granting us access to their data and observations (Redeker and van der Vliet, 2015).

3 Datasets with disagreements

Given the known complexities and disagreements in RST annotations, it has become standard practice in corpus design to include at least a small subset of texts annotated independently by multiple annotators, facilitating measurement of inter-annotator agreement. However, there are substantial differences in how many documents receive parallel annotations, how many discourse units these documents include, and how many annotators are involved. These differences have implications for how helpful the annotations are for learning from disagreement: although the amount of suitable data remains the most important factor, it is certainly not the only one.

Despite this common practice, some datasets lack parallel annotations. Specifically, the Georgetown University Multilayer Corpus (Zeldes, 2017), currently the largest RST treebank, used a development procedure that purposefully avoids measuring the relative annotation quality; as a result, the corpus does not have parallel markup[2]. The Basque RST treebank did not have parallel annotations on the level of whole documents, as its developers measured disagreement on granular tasks, such as the assignment of causal relations (Iruskieta et al., 2013); aside from that, only reconciled annotations are available in the public release. For several corpora, there exist a number of parallel annotations, but these have not been made publicly available for various reasons. This applies to the Potsdam Commentary Corpus (Stede and Neumann, 2014) and APA RST (Hewett, 2023).

Some resources are offered by the RST Discourse Treebank (Lynn Carlson, 2002), formerly the largest RST dataset, containing 385 newswire texts from the Wall Street Journal section of the Penn Treebank. Fifty-three texts from this main corpus body received parallel annotations, providing a relatively large set of parallel RST structures that was published with the main corpus. Still, some

[2] Secondary edges from eRST graphs cannot be fully considered as such, since, for instance, they are not independent from primary ones.

Corpus	N annotators	N docs	N EDUs	Notes
Dutch RST	3	80	2344	Docs unevenly split: 80 / 74 / 13
Kobalt RST	2	42	2216	
CSTNews 6.0	2	5	97	3 or 4 versions for some docs.
Russan RST	3	3	225	
APA RST	3	36	-	*Non-public
RST DT	-	52	2938	*Non-attributed
Spanish RST	-	80	694	*Non-attributed

Table 1: Parallel data in RST corpora. N EDUs assumes the gold standard segmentation.

factors limit the utility of this data for analyzing disagreement.

- Firstly, the primary corpus annotations are not independent of the parallel annotations, as the former result from a reconciliation process involving these parallel versions.

- Secondly, annotations are not explicitly attributed to individual experts, limiting the analysis of annotator-specific perspectives or biases.

The Spanish RST treebank shares the latter two issues, although it remains one of the largest sources in terms of parallel texts, comprising around 700 discourse segments distributed across 80 parallel documents.

For a number of RST treebanks, the opposite is true, i.e., the data is attributed and produced by workers independently, but its amount is insufficient to conduct a feasible quantitative analysis. Such is the case with the Brazilian (CSTNews 6.0, Cardoso et al., 2011) and Russian treebanks (Toldova et al., 2017). We provide the number of annotated documents for these and other corpora in Table 1.

Finally, several corpora feature substantial amounts of attributed parallel annotations, though these are not publicly available and must be requested directly from their creators. A notable example is the Dutch Discourse Treebank (NLDT), which offers three annotation versions for each of its 80 documents (comprising 2,344 EDUs). Typically, two experts annotated each text independently (with a third annotator occasionally participating), followed by a reconciled version (van der Vliet et al., 2011; Redeker et al., 2012). For our analysis, we selected 74 texts annotated by the two experts responsible for the largest annotation share. Although the annotations are not anonymized, for the purpose of our study, we treat the annotators anonymously, labeling them experts A, B, and C.

Another corpus with the desired properties is Kobalt RST (Wan, 2021), a subset of the Kobalt corpus annotated with discourse trees. Similarly, its 42 documents (comprising 2216 EDUs) have three versions: two readings by experts and a reconciliation. Although Kobalt covers a very specific genre of discourse, i.e., argumentative essays by non-native German speakers, it remains suitable for analyzing RST disagreements, such as eliciting individual biases of annotators. We do not incorporate the reconciled annotations in our experiments, as we aim to preserve the raw perspective of each annotator.

Remarkably, both Kobalt and NLDT were annotated by trained experts holding at least a master's degree in linguistics or related disciplines. This expertise level (see Das et al. 2017) and their higher motivation as opposed to crowd annotators ensure the quality of their work. Another similarity is that Kobalt and NLDT concern related languages allowing for a cross-language comparison (which, however, has to account for lexical and syntactic differences). These similarities are another reason why we use both Kobalt and NLDT in our further analysis.

4 An analysis of disagreements in the datasets

In this section, we compare disagreements across the two corpora more closely by reporting confusion matrices and inspecting the label pairs where annotators show consistent divergences. To avoid dealing with matrices that are too nuanced and sparse, we only accounted for cases of disagreement on relations and disregarded cases where experts agree on a relation but disagree on nuclearity[3]. We pay special attention to whether the experts' markup exhibits systematic disagreements. To that end, we consider the most frequently confused relations, dividing them into two categories: "symmetrical" cases, in which annotators A and B confuse relations X and Y equally frequently or at least similarly often, and "asymmetrical" cases, where confusing X and Y is only typical for annotator A or B.

In the first category, we note several tendencies: firstly, problematic relation pairs often involve the ELABORATION relation. Although the annotation manuals for Kobalt and NLDT, the former based on PCC (Stede and Neumann, 2014), treat it differently, it still remains a frequent option that experts resort to when unable to assign a more precise label. While the notion of this relation being problematic has been around for a long time, it is even more evident in a cross-lingual comparison on attributed material. Of more interest is that CAUSE in Kobalt is often confused with other relations by both annotators, sometimes multinuclear and non-causal (LIST). Inspecting the data instances manually, we notice that 81% of these are lexically unspecified and involve adjacent sentences, as in (1).

(1) [Überregionale Produkte werden so stark wie nie konsumiert .] $\xrightarrow{\text{CAUSE/LIST}}$ [Die heutige Generation profitiert von einem vielfältigen Warenangebot dank der Globalisierung ... .]$_{\text{Kobalt\_DEU\_004}}$

Understandably, in this setting, experts struggle to agree on the relative importance of sentences, since normal heuristics, like the deletion test[4], are harder to apply. Likewise, the causality of the relation is also debatable, as human opinions on whether one statement entails another can diverge greatly, as shown by other text understanding tasks (Nie et al., 2020). Some other prominent disagreements, such as those involving JUSTIFY and MOTIVATION in NLDT (Redeker and van der Vliet, 2015), also occur in this underspecified setting.

We report the most frequent disagreements from the second category in Table 2 & Table 3. One of the tendencies we find remarkable is the great number of disagreements over multinuclear relations. This could offer insight into the high value of length as a feature, since multinuclear relations, especially JOINT, which can be used to link arbitrary parts of text, tend to occur in an intersentential position and thus their respective spans are longer in length. Based on the provided numbers, it can be argued that annotators tend to develop a preferred reading for ambiguous cases and assign a specific label based on past experiences. Such is the case with NONVOLITIONAL CAUSE from NLDT, which annotator A considers applicable to a wider range of situations: overall, in our subset of corpus data, expert A uses NONVOLITIONAL CAUSE 111 times, while expert B only 86. Other relations with a similar skew are BACKGROUND (40 vs. 23), JOINT (57 vs. 27), and, to a lesser extent, CONJUNCTION (231 vs. 262).

Incidentally, some of the confusions we observe in Kobalt are also characteristic of other RST corpora: da Cunha et al., 2011 report CONCESSION and ANTITHESIS to be frequently confused in the Spanish treebank. On the other hand, unlike Spanish RST, MEANS and CIRCUMSTANCE are almost never confused in the two corpora, suggesting that the authors' explanation based on connective polysemy is correct.

Relations	Ann. A	Ann. B
conjunction-list	**26**	2
joint-list	2	**7**
concession-antithesis	**8**	0

Table 2: Frequent preferences in Kobalt

Relations	Ann. A	Ann. B
joint-conjunction	1	**22**
nonvol-cause-nonvol-res	**12**	3
list-joint	**11**	1
summary-preparation	**8**	1
nonvol-cause-circumstance	**7**	1

Table 3: Frequent preferences in NLDT

4.1 Results: discussion

The previously made observations shed some light on how various cases of disagreement are distributed in the corpora; we argue that a significant part of these does not constitute an informative signal. One example of this is ELABORATION: keeping this label as an alternative to more specific relations may not be particularly helpful for

[3] We report the most frequently confused relations in the appendix in Table 5 & Table 6.

[4] The deletion test involves removing each part of a relation in turn to determine whether the entire span would retain its original meaning. The part that is harder to delete is considered more important.

understanding the text by either human or machine readers, since the more specific relation often implies that one discourse unit elaborates on the other. Preserving ELABORATION may also have undesired effects during parser training, as parsers tend to develop a bias towards it as the most frequent relation. A further example is constituted by relation types that experts subjectively prefer — possibly, contrary to annotation rules. For instance, the confusion between CONJUNCTION and LIST observed in Kobalt may be a case of this, as the respective manual suggests that LIST should only be assigned when lexical or graphic signals explicitly indicate an enumeration. In cases like that, only one annotator is "correct" with respect to the manual.

However, there also remain plausible divergences in the analyses that can prove informative if preserved in the annotation, such as the CAUSE/LIST example above. The factors behind cases like that include both conflicting or ambiguous signals (several DMs etc.) and underspecification; the latter leads to conflicting readings especially frequently (as another example, consider MOTIVATION and JUSTIFY in NLDT).

In order to determine the more suitable strategy for preserving the meaningful disagreements, it is essential to consider the relative impact of these factors. In the following section, we propose a computational experiment for that purpose.

5 Modeling disagreements

5.1 Motivation

Our experiment aims to quantify the relative impact of surface variables on annotator disagreement, particularly, on discourse relations. In order to do so, we train a classifier for a binary objective: whether two annotators agree or disagree on the relation class given two related discourse units. Our assumption is that signals that consistently prompt diverging interpretations will emerge as important features, while irrelevant signals will not make an impact. To that end, we pick XGBoost as a classifier model that can leverage feature combinations and robustly estimate their contribution (Chen and Guestrin, 2016). As an example, Liu et al., 2023 and Pastor and Oostdijk, 2024 both used XGBoost to analyze hard and easy signals in RST parsing. We also consulted both of these works when determining the set of features.

5.2 Enhancing datasets

For our experiments, we ensured that both corpora were annotated for relevant syntactic and discourse variables, such as UD tags and discourse markers. This required additional intermediate steps as described below.

Concerning syntactic features, we addressed the problem of dependency tagset mismatch. For NLDT, syntactic dependency markup using the Universal Dependencies (UD) standard was published in 2023 as part of the DisRPT shared task (Braud et al., 2023). In contrast, the dependency annotations available for Kobalt use the Hamburg Dependency Treebank (HDT, Borges Völker et al., 2019) annotation standard, which, aside from different tags, also displays a number of differences in tree-building rules (Shadrova, 2020). To ensure that both of our models used syntactic features of similar granularity, we converted the existing dependency annotations from the HDT to the UD standard using a robust converter developed by (Hennig and Köhn, 2017) and obtained standard CONLL-U files.

Discourse features presented a different challenge, namely, the need for a uniform way of annotating both datasets with discourse markers. The task of detecting and disambiguating discourse connectives has drawn significant attention in the context of PDTB-style discourse parsing, with several tools developed specifically for these tasks (Dipper and Stede, 2006; Bourgonje and Stede, 2020). However, these tools only target German and lack a Dutch counterpart. Another development in this direction is the creation of discourse connective inventories for both languages: DimLex (Stede and Umbach, 2002) and DisCoDict (Bourgonje et al., 2018), in which all entries are additionally annotated for possible non-connective readings.

In our approach, we leveraged natural language instructions and used OpenAI's text-to-text generative model O1-mini (OpenAI, 2023) to highlight DM candidates. We purposefully based the model's instructions on a relaxed definition of discourse markers (compared to PDTB), synthesized from Fraser, 2009's account. Our motivation was to cover the entirety of discourse marker candidates to assess their impact on experimental results. The respective prompts are provided in Section A in the appendix.

In the absence of gold DM annotations, we tested the efficiency of this solution using a rule-based baseline that, while imperfect on its own, provides a reliable approximation of ground truth. Specifically, this baseline highlights all entries from DimLex or DisCoDict in the text using regular expressions; however, we discard all matches except those that occur at an EDU-initial position (assuming the existing EDU segmentation). This choice is based on the understanding that a large portion of DM candidates, such as “und” or “en” (“and”) or “als” (“when”), occur at the start of a clausal EDU when acting as subordinating conjunctions and, consequently, as discourse connectives.

We then tested O1-mini’s robustness in detecting these EDU-initial DM candidates, resulting in accuracy scores of 79% and 83% on Kobalt and NLDT, respectively. This, along with a manual inspection we conducted, demonstrates that both O1-mini’s predictions and the baseline show reasonable reliability.

Regarding sources of errors, we note that a large portion of misclassifications occurs due to GPT selecting markers that do not fall into the definition of a discourse connective in PDTB terms and are thus absent from the lexicons we used. These alleged false positives include instances such as “gelukkig” (“luckily”) or “overigens” (“besides”); whether these can truly be regarded as connectives remains an open question.

5.3 Predicting disagreements

Similarly to Liu et al., 2023 and Pastor and Oostdijk, 2024, we do not train the classification algorithm on the text of the two discourse units but only supply it with pre-extracted features. Originally, the features we use were found to be related to item difficulty and could, thus, help predict disagreements; we supply the full list below:

- Discourse unit length in symbols;
- Number of discourse markers (`dm_count`), type of the head DM, i.e., a DM that is the highest in the constituent hierarchy of the second span (`dm`);
- Dependency function of a discourse unit’s syntactic head (`DEPREL` of the head in CONLL-U terms);
- Number of elementary discourse units (roughly, number of clauses) in the first and the second discourse unit, and in total;
- Genre, when applicable;
- Intra-, inter- (involving two sentences), or multisentential status of the relation (Redeker and van der Vliet, 2014) as three binary features;
- Lastly, the label assigned by one of the two annotators, which helps understand whether the experts are in two minds over some particular relation types.

As in the parser-oriented study (Liu et al., 2023), we split our features into two groups. The first group comprises surface features that experts can utilize when annotating a text, while the second group includes the full set of features. The surface feature group includes the following attributes: DU length, the number and type of discourse markers, the syntactic function of the head, and the inter-, intra-, or multisentential status.

Dataset	All	Surface
Kobalt	0.75	0.73
NLDT	0.68	0.59

Table 4: Mean F1 score of XGBoost (5-fold CV)

For each dataset, we separately utilize two subsets of features: surface-only features ("realistic") and all features. We report the average F1 score across a 5-fold cross-validation in Table 4 and provide the relative weights for all factors in Figure 2. It can be seen that, in general, the classifier does not attain an optimal score, especially on NLDT, where the model based on surface features performs slightly above chance. This may indicate that the collected features are insufficient or, at least, do not correlate well with disagreement in NLDT.

5.4 Results: discussion

Despite different classification scores, the two models exhibit a clear pattern in terms of the features they select as relevant. Concretely, discourse unit length always emerges as the most important factor. When the "label" feature is included, it is always the next deciding factor, suggesting that annotators consistently disagree over specific relations: e.g., one picks CAUSE while another picks EXPLANATION. Lastly, the head’s syntactic function and DM type also make a contribution in all settings, although their role in Kobalt seems to be more prominent. Importantly, DM variable appears not as informative as other factors[5].

[5] Evidence from PDTB annotation also demonstrates that agreement does not hinge on the presence of markers: inter-

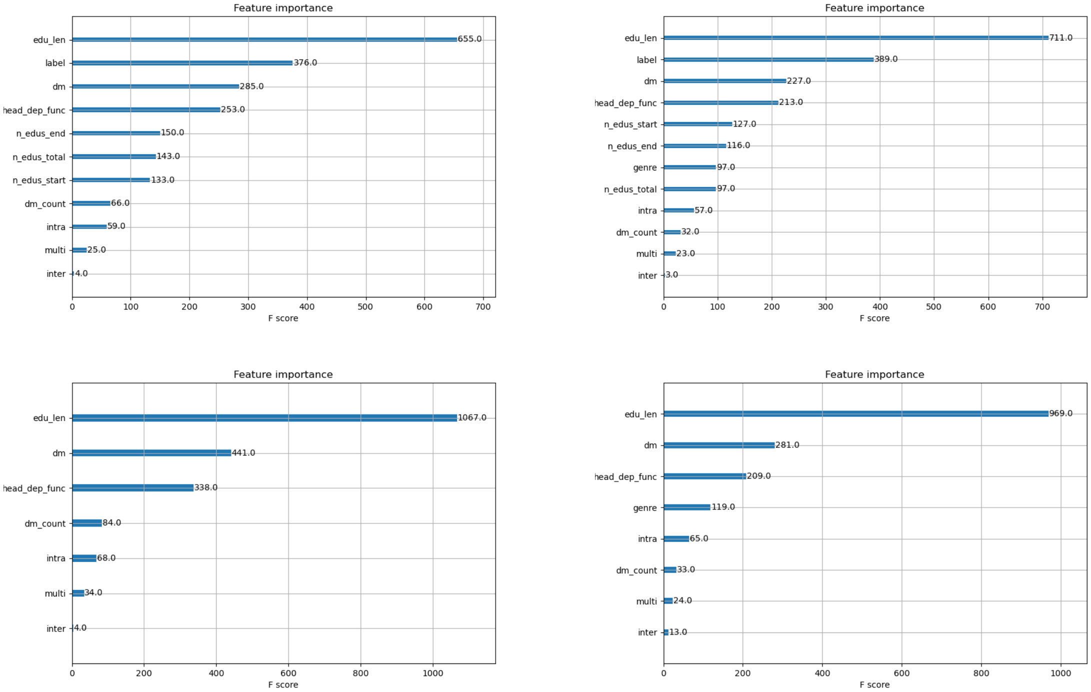


Figure 2: XGBoost weights for Kobalt (left) and NLDT (right): all (top) and surface-only (bottom) features. Abbreviations: edu_len: discourse unit length; dm: discourse marker (present/absent); dm_count: discourse marker count; intra/inter/multi: intra-, inter-, or multisentential relation span; n_edus_start/end: number of EDUs within the first/second argument of the relation.

The first notion aligns well with some of the existing hypotheses about automatic discourse parsing, namely, that humans and parsers struggle more when analyzing relations between lengthy spans of text, as in Nguyen et al., 2021; Shi et al., 2020. Nevertheless, unit length proves to be consistently more important than similar features that account for syntax or tree position: longer spans are often multisentential and include more elementary discourse units, but these factors do not emerge as important.

6 Discussion

The results of our analysis allow us to speculate about the best way of preserving meaningful RST interpretations. As mentioned in Section 2, the two existing alternatives are URML (Das et al., 2017) and eRST (Zeldes et al., 2024); the former of these two could incorporate all parallel readings, and the latter only those that are lexically grounded, i.e., based on one or two discourse markers. Here, we would like to address two properties of eRST annotation that make it less feasible for this task.

The first of these is its definition of discourse markers, which serve as a basis for secondary edges. In this respect, eRST aligns completely with PDTB's notion of discourse connectives and its respective restrictions: only subordinating conjunctions, coordinating conjunctions, and adverbials can have the status of discourse markers (Zeldes et al., 2024). In this paper, we are not looking to contribute to the vast theoretical discussion on what lexical elements should be considered discourse markers; however, we must note that existing studies offer different answers to this question, sometimes using the same linguistic material. For instance, annotating the Wall Street Journal corpus with PDTB-style discourse connectives (PDTB 2.0, Prasad et al., 2008) and with more vaguely defined discourse markers (RST Signalling Corpus, Das and Taboada, 2017; Das, 2014) results in a different number of unique markers being identified: 100 and 201, respectively. Partly, this is due to the latter category including combinations like "but also", but also due to inclusion of broader lexical

annotator agreement for implicit relations (85.1%, Prasad et al., 2008) is only slightly lower than for explicit ones (90.2%).

categories.

Undoubtedly, adopting a stricter definition simplifies the task for corpus annotators, resulting in better reliability of their work. On the other hand, it raises the question of whether using a broader set of markers, such as that of the RST Signalling Corpus, would allow for broader coverage of secondary edges and better reflect the space of possible interpretations of discourse—something that eRST, as well as ourselves, seeks to address. For example, such items as "naturally", "of course", and "after all" are not listed as explicit in either PDTB 2.0 (Prasad et al., 2008) or PDTB 3.0 (Prasad et al., 2019). However, we could model cases where "naturally" would signal REASON relation and "after all" would signal CAUSE. In eRST terms, it would prompt the addition of a primary or a secondary edge.

(2) [We only left home at 8;]$\xleftarrow{\text{REASON}}$[naturally, we were late.]

(3) [He will do that for you,]$\xrightarrow{\text{CAUSE}}$[because, after all, he is your brother.]

These examples suggest that relaxing the existing lexical criteria for secondary edges could, in theory, improve coverage.

A further possible shortcoming of eRST is that it cannot incorporate plausible readings of underspecified relations unlike URML. This is especially important since in the existing corpora, the larger part of relations is not signalled by markers (Taboada, 2006; Das and Taboada, 2017). Our observations also confirm that disagreement is strongly associated with underspecification; thus, we argue that a standard that aims to integrate parallel readings will profit from allowing multiple graph edges in underspecified cases.

7 Conclusion

The analyses presented in this paper highlight that RST annotations exhibit a persistent and systematic degree of inter-annotator disagreement. Drawing on two expert-annotated corpora (Dutch and German), we observe that divergent interpretations often arise from the inherent complexity of discourse relations, especially when label definitions are underspecified or conflated. Although some discrepancies reflect an annotator's systematic bias (e.g., favoring ELABORATION or LIST), in many cases, multiple readings of a relation are equally plausible. Our experiments suggest that span length and certain label choices serve as strong predictors of disagreement, indicating that large or complex discourse spans are particularly prone to ambiguous interpretations.

From an applied perspective, two complementary strategies emerge. First, filtering out demonstrable biases that run counter to annotation rules can clarify the "true" consensus. Here, the judgment needs to be based around surface signals handled differently than prescribed; consequently, even rule-based systems or simpler neural language models can prove helpful at this task.

Second, adopting flexible schemes that capture legitimate ambiguity, such as URML or eRST, can more comprehensively reflect discourse complexity; of these two, we find URML better suited for this (and only for this) specific task, as it gives more freedom for genuine discrepancies to be integrated. Moving forward, these dual approaches — tightening clearly defined guidelines while embracing multiple valid analyses — hold promise for improving both the reliability and the expressive power of RST annotation.

Limitations

We acknowledge that our analysis focuses on RST relations paying less attention to the partly overlapping problems of disagreements in nuclearity and discourse unit spans. Furthermore, we highlight that the features we used when predicting disagreement do not offer an exhaustive picture of factors behind annotation discrepancies. Considering additional variables, such as rhetorical "moves" (Redeker et al., 2012) or syntactic signals beyond clause boundaries, could make the analysis more complete.

Acknowledgments

We thank the anonymous reviewers, Hugh Mee Wong, Frances Yung, and the NLP group at the University of Utrecht for their helpful feedback on this work. We are thankful to Gisela Redeker and Gosse Bouma for granting us access to the early versions of the Dutch discourse treebank. This work is funded by NWO through an AINed Fellowship Grant NGF.1607.22.002 and supported by project 'Dealing with Meaning Variation in NLP'.

References

Sohail Akhtar, Valerio Basile, and Viviana Patti. 2020. Modeling annotator perspective and polarized opinions to improve hate speech detection. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 8, pages 151–154.

Lora Aroyo and Chris Welty. 2014. The three sides of crowdtruth. *Hum. Comput.*, 1:31–44.

N. Asher and A. Lascarides. 2003. *Logics of Conversation*. Studies in Natural Language Processing. Cambridge University Press.

Emanuel Borges Völker, Maximilian Wendt, Felix Hennig, and Arne Köhn. 2019. HDT-UD: A very large Universal Dependencies treebank for German. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 46–57, Paris, France. Association for Computational Linguistics.

Peter Bourgonje, Jet Hoek, Jacqueline Evers-Vermeul, Gisela Redeker, Ted J M Sanders, and Manfred Stede. 2018. Constructing a lexicon of dutch discourse connectives.

Peter Bourgonje and Manfred Stede. 2020. Exploiting a lexical resource for discourse connective disambiguation in german. In *International Conference on Computational Linguistics*.

Chloé Braud, Yang Janet Liu, Eleni Metheniti, Philippe Muller, Laura Rivière, Attapol Rutherford, and Amir Zeldes. 2023. The DISRPT 2023 shared task on elementary discourse unit segmentation, connective detection, and relation classification. In *Proceedings of the 3rd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2023)*, pages 1–21, Toronto, Canada. The Association for Computational Linguistics.

Paula Christina Figueira Cardoso, Erick Galani Maziero, Mara Elena Lucia, R. Castro Jorge, Ariani Di Felippo, Lucia Helena Machado Rino, Maria das Graças, Volpe Nunes, Thiago Alexandre Salgueiro Pardo, and Rodovia Washington Luís. 2011. Cstnews - a discourse-annotated corpus for single and multi-document summarization of news texts in brazilian portuguese.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Iria da Cunha, Juan-Manuel Torres-Moreno, and Gerardo Sierra. 2011. On the development of the rst spanish treebank. In *Proceedings of the 5th Linguistic Annotation Workshop*, LAW V '11, pages 1–10, Stroudsburg, PA, USA. Association for Computational Linguistics.

Debopam Das. 2014. *Signalling of Coherence Relations in Discourse*. Ph.D. thesis, Simon Fraser University.

Debopam Das, Manfred Stede, and Maite Taboada. 2017. The good, the bad, and the disagreement: Complex ground truth in rhetorical structure analysis. In *Proceedings of the 6th Workshop on Recent Advances in RST and Related Formalisms*, pages 11–19, Santiago de Compostela, Spain. Association for Computational Linguistics.

Debopam Das and Maite Taboada. 2017. Rst signalling corpus: a corpus of signals of coherence relations. *Language Resources and Evaluation*, 52:149 – 184.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Stefanie Dipper and Manfred Stede. 2006. Disambiguating potential connectives.

Tommaso Fornaciari, Alexandra Uma, Silviu Paun, Barbara Plank, Dirk Hovy, and Massimo Poesio. 2021. Beyond black & white: Leveraging annotator disagreement via soft-label multi-task learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2591–2597, Online. Association for Computational Linguistics.

Bruce L. Fraser. 2009. An account of discourse markers. *International Review of Pragmatics*, 1:293–320.

Felix Hennig and Arne Köhn. 2017. Dependency tree transformation with tree transducers. In *UDW@NoDaLiDa*.

Freya Hewett. 2023. APA-RST: A text simplification corpus with RST annotations. In *Proceedings of the 4th Workshop on Computational Approaches to Discourse (CODI 2023)*, pages 173–179, Toronto, Canada. Association for Computational Linguistics.

Mikel Iruskieta, María Jesús Aranzabe, Arantza Díaz de Ilarraza, Itziar Gonzalez-Dios, Mikel Lersundi, and Oier Lopez de Lacalle. 2013. The rst basque treebank : an online search interface to check rhetorical relations.

Mikel Iruskieta, Iria da Cunha, and Maite Taboada. 2015. A qualitative comparison method for rhetorical structures: identifying different discourse structures in multilingual corpora. *Language Resources and Evaluation*, 49(2):263–309.

Yang Janet Liu, Tatsuya Aoyama, and Amir Zeldes. 2023. What's hard in english rst parsing? predictive models for error analysis. *Preprint*, arXiv:2309.04940.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Mary Ellen Okurowski Lynn Carlson, Daniel Marcu. 2002. *RST Discourse Treebank*. Philadelphia: Linguistic Data Consortium.

Aru Maekawa, Tsutomu Hirao, Hidetaka Kamigaito, and Manabu Okumura. 2024. Can we obtain significant success in rst discourse parsing by using large language models? *Preprint*, arXiv:2403.05065.

William C Mann and Sandra A Thompson. 1988. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Daniel Marcu. 1996. Building up rhetorical structure trees. In *Proceedings of AAAI-96*, pages 1069–1074, Portland, OR.

Daniel Marcu. 2000. *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press.

Johanna D. Moore and Martha E. Pollack. 1992. A problem for RST: The need for multi-level discourse analysis. *Computational Linguistics*, 18(4):537–544.

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110.

Thanh-Tung Nguyen, Xuan-Phi Nguyen, Shafiq Joty, and Xiaoli Li. 2021. RST parsing from scratch. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1613–1625, Online. Association for Computational Linguistics.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Martial Pastor and Nelleke Oostdijk. 2024. Signals as features: Predicting error/success in rhetorical structure parsing. In *Proceedings of the 5th Workshop on Computational Approaches to Discourse (CODI 2024)*, pages 139–148, St. Julians, Malta. Association for Computational Linguistics.

Joshua C. Peterson, Ruairidh M. Battleday, Thomas L. Griffiths, and Olga Russakovsky. 2019. Human uncertainty makes classification more robust. *CoRR*, abs/1908.07086.

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014. Learning part-of-speech taggers with inter-annotator agreement loss. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 742–751, Gothenburg, Sweden. Association for Computational Linguistics.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Rashmi Prasad, Bonnie Webber, Alan Lee, and Aravind Joshi. 2019. *Penn Discourse Treebank Version 3.0*. Abacus Data Network.

Gisela Redeker, Ildikó Berzlánovich, Nynke van der Vliet, Gosse Bouma, and Markus Egg. 2012. Multi-layer discourse annotation of a dutch text corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2820–2825, Istanbul, Turkey. European Language Resources Association (ELRA).

Gisela Redeker and Nynke van der Vliet. 2014. Explicit and implicit coherence relations in dutch texts. *Pragmatics and beyond. New series*, 254:23–52.

Gisela Redeker and Nynke van der Vliet. 2015. Exploring and evaluating rst annotations. *Unpublished manuscript*.

David Reitter and Manfred Stede. 2003. Step by step: underspecified markup in incremental rhetorical analysis. In *Proceedings of 4th International Workshop on Linguistically Interpreted Corpora (LINC-03) at EACL 2003*.

Filipe Rodrigues and Francisco Pereira. 2017. Deep learning from crowds. *Preprint*, arXiv:1709.01779.

T.J.M. Sanders and W.P.M.S. Spooren. 1999. *Communicative intentions and coherence relations*, pages 235–250. Number 63 in Pragmaticsamp;Beyond. J. Benjamins.

Merel C. J. Scholman and Vera Demberg. 2017. Crowdsourcing discourse interpretations: On the influence of context and the reliability of a connective insertion task. In *LAW@ACL*.

Anna Valer'evna Shadrova. 2020. *Measuring coselectional constraint in learner corpora: A graph-based approach*. Ph.D. thesis, Humboldt-UniversitÃ¤t zu Berlin, Sprach- und literaturwissenschaftliche FakultÃ¤t.

Victor S. Sheng, Foster J. Provost, and Panagiotis G. Ipeirotis. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. *Organizations & Markets eJournal*.

Ke Shi, Zhengyuan Liu, and Nancy F. Chen. 2020. An end-to-end document-level neural discourse parser exploiting multi-granularity representations. *CoRR*, abs/2012.11169.

Manfred Stede. 2008a. Disambiguating rhetorical structure. *Research on Language and Computation*, 6:311–332.

Manfred Stede. 2008b. Rst revisited : disentangling nuclearity.

Manfred Stede and Arne Neumann. 2014. Potsdam commentary corpus 2.0: Annotation for discourse research. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 925–929, Reykjavik, Iceland. European Language Resources Association (ELRA).

Manfred Stede and Carla Umbach. 2002. Dimlex: A lexicon of discourse markers for text generation and understanding. In *International Conference on Computational Linguistics*.

Maite Taboada. 2006. Discourse markers as signals (or not) of rhetorical relations. *Journal of Pragmatics*, 38(4):567–592. Focus-on Issue: The Pragmatics of Discourse Management.

Maite Taboada and William C. Mann. 2006. Rhetorical structure theory: looking back and moving ahead. *Discourse Studies*, 8:423 – 459.

Svetlana Toldova, Dina Pisarevskaya, M. I. Ananyeva, Maria Kobozeva, Alexandr Nasedkin, S. Nikiforova, Irina Petrovna Pavlova, and Alexey Shelepov. 2017. Rhetorical relations markers in russian rst treebank.

Alexandra Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2020. A case for soft loss functions. In *AAAI Conference on Human Computation & Crowdsourcing*.

N.H. van der Vliet, I. Berzlánovich, G. Bouma, M. Egg, and G. Redeker. 2011. Building a discourse-annotated dutch text corpus. In *Beyond Semantics*, volume 3 of *Bochumer Linguistische Arbeitsberichte*, pages 157 – 171. Ruhr-Universität Bochum.

Shujun Wan. 2021. Kobalt_rst (rst german learner treebank): die annotation von rhetorischen strukturen im kobalt-daf-korpus.

Shujun Wan, Tino Kutschbach, Anke Lüdeling, and Manfred Stede. 2019. RST-tace a tool for automatic comparison and evaluation of RST trees. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 88–96, Minneapolis, MN. Association for Computational Linguistics.

Frances Yung, Merel C. J. Scholman, Sárka Zikánová, and Vera Demberg. 2024. Discogem 2.0: A parallel corpus of english, german, french and czech implicit discourse relations. In *International Conference on Language Resources and Evaluation*.

Amir Zeldes. 2016. rstWeb - a browser-based annotation interface for Rhetorical Structure Theory and discourse relations. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 1–5, San Diego, California. Association for Computational Linguistics.

Amir Zeldes. 2017. The gum corpus: creating multilayer resources in the classroom. *Lang. Resour. Eval.*, 51(3):581–612.

Amir Zeldes, Tatsuya Aoyama, Yang Janet Liu, Siyao Peng, Debopam Das, and Luke Gessler. 2024. erst: A signaled graph theory of discourse relations and organization. *Computational Linguistics*, pages 1–47.

A Connective detection prompts

A.1 NLDT connective detection prompt

Instruction
In the following Dutch text, identify all discourse markers (DMs) and enclose them in <dm> tags.
Definition of Discourse Markers (DMs):

- DMs, also known as connectives, are lexical expressions (e.g., *en*, *maar*, *omdat*, *dus*, *hoewel*, *toch*) that belong to different syntactic classes such as conjunctions, adverbials, and prepositional phrases.

- They are used to connect discourse components (text segments) and signal the coherence relations that hold between those components (e.g., contrast, cause, elaboration).

- The scope of a DM's function is a single discourse sequence comprising adjacent text spans in a relation.

- DMs can be present at the beginning, middle, or end of a sentence (or segment).

- A DM signals relations that hold between two adjacent text segments but does not create the relation; it guides the interpretation of the relation.
Guidelines:

1. **Scope of DMs:**

- The function of a discourse marker applies to a single discourse sequence comprising adjacent text spans in a relation.

- DMs signal relations that hold between two adjacent text segments.

- A discourse marker does not create the relation between text segments; it only guides the interpretation of the relation.

2. **Position of DMs:**

- DMs can be present at the beginning, middle, or end of a sentence (or segment).

- They may appear within the sentence or at clause boundaries.

3. **Identification of DMs:**

- Use a list of common Dutch DMs to identify potential markers, such as:

- **Addition:** *en*, *ook*, *bovendien*

- **Contrast:** *maar*, *echter*, *toch*

- **Condition:** *als*, *indien*, *tenzij*

- **Cause/Reason:** *omdat*, *want*, *doordat*

- **Concession:** *hoewel*, *ofschoon*, *desondanks*

- **Temporal:** *toen*, *terwijl*, *voordat*, *nadat*

- **Result/Consequence:** *dus*, *daardoor*, *zodat*

- **Example:** *bijvoorbeeld*, *zoals*

- Ensure the word functions as a DM in context by connecting two propositions or clauses.

- Confirm that the token's part of speech corresponds to typical DM categories (conjunctions, adverbials, prepositional phrases).

4. **Annotation Format:**

- Enclose each identified DM within <dm> and </dm> tags.

- Do not alter the original text other than adding the tags around the DMs.

5. **Examples:**

English Example:

Input:

"A country is considered financially healthy **if** its reserves cover three months of its imports."

Output:

"A country is considered financially healthy <dm>if</dm> its reserves cover three months of its imports."

Dutch Examples:

Example 1:

Input:

"Drie nieuwe emissies beginnen vandaag te handelen op de New York Stock Exchange, **en** één begon vorige week te handelen op de Nasdaq/National Market System."

Output:

"Drie nieuwe emissies beginnen vandaag te handelen op de New York Stock Exchange, <dm>en</dm> één begon vorige week te handelen op de Nasdaq/National Market System."

Example 2:

Input:

"De Poolse rat zal deze winter goed eten. Tonnen heerlijk rottende aardappelen, gerst en tarwe zullen vochtige schuren over het hele land vullen **terwijl** duizenden boeren de kopers van de staat wegsturen."

Output:

"De Poolse rat zal deze winter goed eten. Tonnen heerlijk rottende aardappelen, gerst en tarwe zullen vochtige schuren over het hele land vullen <dm>terwijl</dm> duizenden boeren de kopers van de staat wegsturen."

Task:

- Read the following Dutch text.

- Identify all discourse markers based on the guidelines above.

- Enclose each DM within <dm> tags.

- Ensure that the rest of the text remains unchanged.

Notes:

- Pay special attention to words that can function as DMs but may have other grammatical roles. Use context to determine their function.

- The goal is to produce a text identical to the input except for the addition of <dm> tags around the identified discourse markers.

- Do not tag words that are not functioning as discourse markers in the given context.

By following these instructions, you will identify and annotate all discourse markers in the text, which will help in analyzing the coherence relations within the text and assist in computational processing.

Text to Process:

A.2 Kobalt connective detection prompt

Instruction

In the following German text, identify all discourse markers (DMs) and enclose them in <dm> tags.

Definition of Discourse Markers (DMs):

- DMs, also known as connectives, are lexical expressions (e.g., und, weil, obwohl) that belong to different syntactic classes such as conjunctions, adverbials, and prepositional phrases.

- They are used to connect discourse components (text segments) and signal the coherence relations that hold between those components (e.g., contrast, cause, elaboration).

- The scope of a DM's function is a single discourse sequence comprising adjacent text spans in a relation.

- DMs can be present at the beginning, middle, or end of a sentence (or segment).

- A DM signals relations that hold between two adjacent text segments but does not create the relation; it guides the interpretation of the relation.

Guidelines:

1. **Identify Potential DMs:**

- **Common DMs in German include:**

- **Conjunctions:** und (and), aber (but), oder (or), denn (for), sondern (but rather), weil (because), obwohl (although), wenn (if), während (while), falls (in case).

- **Adverbials:** deshalb (therefore), trotzdem (nevertheless), allerdings (however), außerdem (besides), folglich (consequently), inzwischen (meanwhile), dennoch (still).

- **Prepositional Phrases:** im Gegensatz zu (in contrast to), aufgrund von (due to), trotz (despite), infolgedessen (as a result).

2. **Position in Sentence:**

- DMs can appear at the beginning, middle, or end of a sentence.

- Examples:

- Initial: <dm>Trotzdem</dm> geht er zur Arbeit. (Nevertheless, he goes to work.)

- Medial: Er geht <dm>trotzdem</dm> zur Arbeit.

- Final: Er geht zur Arbeit, <dm>trotzdem</dm>.

3. **Confirm the Function:**

- Ensure the word or phrase is functioning as a DM and not in another grammatical role.

- Exclude words that are not functioning as DMs (e.g., "dass" as a complementizer). Exclude "dass" as a complementizer. Exclude "dass" as a complementizer.

- Exclude "dass" as a complementizer.

- Exclude "und" if not interclausal.

Examples:

1. **Example (English DMs):**

- **Relation DMs:**

- Circumstance: when, as, with

- Condition: if, unless

- Contrast: but, however

- Concession: while, though

- Elaboration-additional: and, also

- Reason: because, due to

- List: and, in addition, moreover

- Temporal-after: since, after

- Temporal-before: before

2. **Example 1:**

Three new issues begin trading on the New York Stock Exchange today, <dm>and</dm> one began trading on the Nasdaq/National Market System last week. On the Big Board, Crawford & Co., Atlanta, (CFD) begins trading today. Crawford evaluates health care plans, manages medical and disability aspects of worker's compensation injuries <dm>and</dm> is involved in claims adjustments for insurance companies. <dm>Also</dm> beginning trading today on the Big Board are El Paso Refinery Limited Partnership, El Paso, Texas, (ELP) and Franklin Multi-Income Trust, San Mateo, Calif., (FMI).

3. **Example 2:**

The Polish rat will eat well this winter. Tons of delectably rotting potatoes, barley and wheat will fill damp barns across the land <dm>as</dm> thousands of farmers turn the state's buyers away. Many a piglet won't be born as a result, <dm>and</dm> many a ham will never hang in a butcher shop. <dm>But</dm> with inflation raging, grain in the barn will still be a safer bet for the private farmer than money in the bank. Once again, the indomitable peasant holds Poland's future in his hands. <dm>Until</dm> his labor can produce a profit in this dying and distorted system, even Solidarity's sympathetic new government won't win him over.

Your Task:

- Read the following German text.

- Identify all DMs as per the guidelines above.

- Enclose each DM within <dm> tags.

- Ensure that the rest of the text remains unchanged.

German Text:

B Frequently confused relations

Relations	Ann. A	Ann. B
elaboration-evidence	13	10
cause-list	7	8
cause-reason	5	5
cause-evidence	6	4
cause-elaboration	5	4
conjunction-list	26	2
joint-list	2	7
concession-antithesis	8	0

Table 5: Frequent two-sided (top) and one-sided (bottom) relation confusions in Kobalt

Relations	Ann. A	Ann. B
elaboration-interpretation	15	11
elaboration-nonvol-cause	15	11
elaboration-circumstance	14	11
elaboration-nonvol-result	12	11
elaboration-background	6	12
elaboration-conjunction	11	6
circumstance-condition	10	5
elaboration-preparation	8	7
justify-motivation	7	5
joint-conjunction	22	1
nonvol-cause-nonvol-res	12	3
joint-list	11	1
summary-preparation	8	1
nonvol-cause-circumstance	7	1

Table 6: Frequent two-sided (top) and one-sided (bottom) relation confusions in NLDT

Aligning Embedding Spaces Across Languages to Identify Word Level Equivalents in the Context of Concreteness and Emotion

Josephine Kaminaga*[1], Jingyi Wu*[1,2], Daniel Yeung*[1], and Simon Todd[1]

[1]University of California, Santa Barbara
{jkaminaga,jingyi_wu,dyeung,sjtodd}@ucsb.edu
[2]Cornell University
{jw2824}@cornell.edu

Abstract

The impact of emotionality and abstraction on language processing has been heavily studied in monolingual and, to an extent, bilingual settings. Most of these studies were experiments with humans that yielded mixed results regarding the exact effect of emotionality or abstraction on cross-linguistic tasks. To elucidate this relationship between translation, emotionality, and abstraction, we used a neural network to model a bilingual mapping within an English-Mandarin semantic space. We sought to understand what our quantitative results implied about structural differences between English and Mandarin lexical semantic spaces. Overall, our model translated concrete and emotion-laden words more accurately than abstract and emotionally neutral words, suggesting that strong concreteness and emotionality are more consistently perceived across languages. On a more detailed level, our model learned clusters of some related groups of words in both languages, but failed to create a 1-to-1 semantic mapping, with several types of errors we hypothesize are due to linguistic and cultural differences. Our results indicate interesting possibilities for using quantitative word-level modeling as a tool to analyze the overlapping impacts of bilingualism, emotionality, and abstraction on each other.

1 Introduction

Emotionality and abstraction have long been important topics of analysis in psycholinguistics. Emotionality is typically measured along the dimensions of valence - the positivity/negativity of a word - and arousal - the level of activation a word inspires, or "the negative probability of falling asleep" (Altarriba and Sutton, 2004). Abstraction is measured through concreteness: the extent to which a word denotes a physical object, action, or property. These measures form a basis for linguistic conceptual spaces and are dimensions along which words are categorized and understood (Altarriba et al., 1999; Altarriba and Bauer, 2004). A significant body of work investigating the role of emotionality and abstraction in the processing and interpretation of words has been produced (Altarriba and Bauer, 2004; Altmann, 2001; Hinojosa et al., 2020; Majid, 2012). It has been shown, for example, that concreteness lends itself to quicker concept acquisition and word processing, (Guasch and Ferré, 2021), that highly emotional words are processed faster than non-emotional ones (Kousta et al., 2011), and that there is a "negative bias" wherein emotionally negative stimuli take longer to process than emotionally positive ones (Bromberek-Dyzman et al., 2021; Mergen and Kuruoglu, 2017). While most of these conclusions were drawn from monolingual studies, it is worthwhile to study how emotionality and abstraction impact word mapping in a bilingual semantic space. How do these dimensions characterize words in each language, and can these characterizations be mapped accurately across languages?

Existing research in this area has shown that increased levels of concreteness confer advantages in monolingual word processing and bilingual word translation (Binder et al., 2005; Guasch and Ferré, 2021; Ferré et al., 2017). These benefits may result from the referents of abstract words having greater ambiguity and variety, and less tactile representations, than concrete words (Pauligk et al., 2019). Emotional valence confers similar processing advantages in monolingual and multilingual contexts (Kousta et al., 2011; Ferré et al., 2017). This is likewise attributed to the constriction of the available referent space, as strong values of emotional valence highlight recognizability of certain concepts (Kousta et al., 2011), which facilitates processing of those concepts' lexical representations. This effect is known to interact with concrete-

*Equal Contribution. Authors listed in alphabetical order.

ness levels, with enhanced effects for more abstract stimuli (Kousta et al., 2011; Altarriba and Bauer, 2004). In summary, words with high valence or concreteness represent concepts with increased recognizability, and confer processing advantages due to their emotional specificity or tactile imageability, respectively. We hypothesize the contexts in which such words are used reflect this. Specifically, there should be more similarity across the contexts in which a concrete word is used, narrower in variation than the contexts of abstract word usage. While some recent research in cross-linguistic semantic alignment has suggested that concreteness is uncorrelated with alignment, it was also found that semantic domains with "high internal coherence" have a "low dimensionality" that "seems to enable high alignment" (Thompson et al., 2020). This finding suggests that the narrower the variation of a given concept's associations, the greater ease of cross linguistic alignment. If this is the case, then our model should perform better on words with narrower contextual variation.

The majority of bilingual studies on this topic have focused on sequential bilinguals and the difference between L1 and L2 processing (Sharif and Mahmood, 2023). The literature on the impact of emotionality and abstraction for bilingual processing has come to widely varied conclusions that disagree based on the study structure and language, the words used to test processing, and even the population discrepancies among studied bilingual communities (Ferré et al., 2017). Given these results, it is reasonable to turn our attention to simultaneous bilinguals. They have learned both languages as L1s, and the L1/L2 discrepancies (e.g. age and context of L2 acquisition, and frequency of L2 usage) that affect processing tasks would likely have less of an impact (Liao and Ni, 2022; Pavlenko, 2012; Ponari et al., 2015). This would create a more even space in which to study cross-language differences in emotionality and abstraction. However, despite acknowledgment that this is a promising direction of study, there are only a handful of papers investigating how simultaneous bilinguals process emotionality and abstraction (Sharif and Mahmood, 2023). Due to the lack of research into simultaneous bilingualism and given the extractable nature of representations in computational modeling, using computational methods to simulate simultaneous bilingual spaces could yield fruitful results.

Computational modeling of language has a long, interdisciplinary history of usage in linguistics and psychology (Grishman, 1989; Krahmer, 2010; Jurafsky and Martin, 2008). It benefits from using a diverse range of language corpora instead of being restricted to participants with highly specific language experience. We postulate that if a model learns the contexts in which words with varying concreteness and emotionality are used across languages, it could mirror the patterns of simultaneous bilingual human participants in cross-linguistic processing tasks, such as interlingual lexical decision tasks or translation pair production tasks. Such a model would yield large amounts of information on how the two dimensions impact word translation and semantic space mapping in a bilingual environment, as the model's outputs would provide explicit access to cross-linguistic representations of words that can be visualized to understand their structure.

Thus, in this paper, we develop a word-level neural network translation model for English and Mandarin Chinese. Given pretrained monolingual embeddings from two languages, our model's goal is to learn a simultaneous semantic mapping between the two languages. While simpler alignment methods, such as Orthogonal Procrustes (Schönemann, 1966), offer a useful baseline for aligning embedding spaces, they assume a strict one-to-one correspondence between words across languages. This assumption does not hold in our setting, where an English word can have multiple valid translations in Chinese depending on context. In contrast, our encoder-decoder model can implicitly learn one-to-many mappings and better capture the complexity of cross-linguistic semantics.

We also considered using more modern architectures, such as Transformer-based models (Vaswani et al., 2017), which are widely used in contemporary neural machine translation. However, Transformer models operate on subword token sequences rather than whole-word embeddings, making their learned representations harder to interpret in terms of cross-lingual semantic structure. Since our goal is to analyze how emotionality and abstraction affect translation at the word level, the encoder-decoder framework offers a more interpretable and semantically meaningful approach.

By testing the model's translation abilities on words with different levels of emotionality and abstraction, we can investigate the impacts of differing emotionality and abstraction on cross-linguistic processing, and analyze the between-language structure of the two dimensions. As we hypothesize the contexts of word use reflect the traits of the concepts they represent, we theorize that our model, through learning such contexts, will have greater translation performance on words with greater emotionality and concreteness levels, reflecting results from prior human studies (Ferré et al., 2017). Our model's results are interpreted in the context of using computational modeling to improve accessibility of further research into two related areas: How emotion and abstraction varies structure between languages, and the bilingual processing of these categories. [1]

2 Methods

2.1 Data

We chose English and Mandarin Chinese as our languages of investigation due to the relatively high accessibility of emotionality/concreteness ratings and corpora for them, as well as the accessibility of simultaneous bilingual participants in the event of a human-participant extension for this study. Our training and testing data consisted of 38,000 pairs of English words and their Chinese translation equivalents. These pairs were sourced from 6 different online English-to-Chinese dictionaries - Cambridge, Yabla, MDBG, Facebook MUSE dataset, ECDICT, and CEDICT (Cambridge, 2024; Yabla; MDBG; Conneau et al., 2017; Lin, 2024; CC-CEDICT). We obtained these pairs by querying each dictionary from a list of 119,354 English words taken from the UNISYN English lexicon, altogether covering a great variety of emotional, abstract, and concrete words. All models in this paper used the pretrained, 200-dimensional English and Chinese embeddings, created by the Tencent AI lab via a bidirectional skip-gram model. To ensure total overlap between the training data and the pretrained embeddings, preprocessing was done on the training data to filter out any pairs that included words not in either set of embeddings.

After obtaining our dataset, it was separated into the three aforementioned classes of words: concrete, abstract, and emotional. This was done by using an online database of 40,000 English words rated on mean concreteness/abstraction in a 5 point scale from 1 (abstract) to 5 (concrete) (Brysbaert et al., 2014). This database was then split into two categories. Words with a lower concreteness rating than the median rating were categorized as abstract, and words with a higher concreteness rating than the median rating were categorized as concrete. Words within 1 point of rating from the median were then categorized as "weak" abstract and concrete words. Words that had exactly 2,3, or 4 as a rating were excluded, as these were the exact points on which we divided our dataset. Emotion words are split into two categories: emotion label words, or words that serve as representations for emotions, and emotion-laden words, words with high emotional values/associations. Using separate databases of 497 emotion label words and 6453 emotion-laden words (Zupan et al., 2023; Mohammad and Turney, 2013), we identified the words in our set that fit into either of these categories to generate our emotion word set. Emotion words are contextualized by their arousal and valence ratings, or how pleasant/unpleasant and how intense a word is. We utilized a dataset of these ratings for 14,000 English lemmas (Warriner et al., 2013) to tag and measure the emotional properties of our emotion words. As many emotion label words, such as "grave", are polysemous with emotion-laden words, we collapse the two categories into a singular emotion word category for the purpose of testing.

We partitioned a lemmatized version of our dataset (lemmatized using NLTK WordNet lemmatizer (Bird et al., 2009)) by comparing every word in these datasets against our list of concrete, absolute, and emotion words. With this, we were able to create a dataset split across the three categories. Each category's data was then split into 10 equal batches, then each batch was linked across categories. This way, we had proportional chunks of the dataset to train or test on that each contained 10 percent of all the concrete, abstract, and emotion words. This was done to ensure each batch more consistently reflected realistic proportions of all categories.

2.2 Latent Space Transformation

A common challenge in training Neural Networks (NNs) is the variability of the learned latent representations, even when the task and data distribution remain fixed. Stochastic factors such as weight initialization, data shuffling, and hyperparameter settings can lead to different latent spaces across training runs (Wang et al., 2018). While these embeddings may vary in their absolute coordinates, they often preserve relative distances and differ only by an isometric transformation. This

[1] There are many types of bilinguals; we assume both of the model's lexicons are stable and well defined, similar to simultaneous bilinguals'. That is, we aimed not to model the acquisition of a lexicon but rather to model the processing behind mapping two fully formed lexicons.

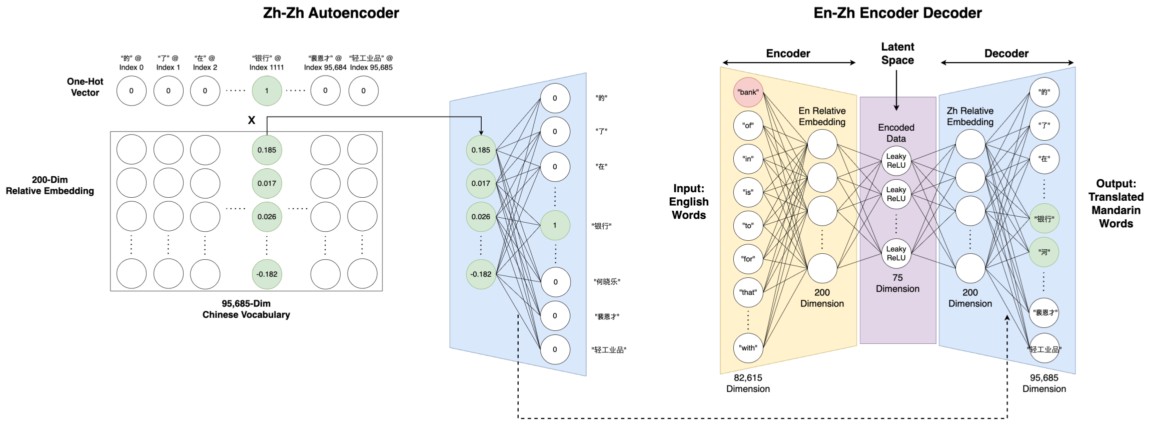


Figure 1: Architecture design of the models in this paper. Note that the trained weights from the Zh-Zh Autoencoder are directly transferred to the decoder in En-Zh Encoder Decoder model, as pointed by the dotted line.

variability complicates tasks like comparing representations across models or reusing pretrained components. To address this, Moschella et al. (Moschella et al., 2022) proposed using relative representations—computed as cosine similarities between selected anchor words and the rest of the vocabulary—to provide a stable, geometry-invariant alternative. By transforming the original latent space to the one represented by relative embeddings, Moschella et al. demonstrated the desired invariance to isometric and scaling transformations, which makes zero-shot stitching of models possible. Adopting the relative embeddings in our models should presumably improve the translation accuracy as the latent spaces for English and Chinese are invariant to the stochastic factors mentioned above and are optimal in encoding the translation information after transformation.

Mathematically, the transformation is achieved as follows. Given a training set $\mathbb{X}$, an embedding function $E_\theta : \mathbb{X} \rightarrow \mathbb{R}^d$ parameterized by θ is learned to map each sample $x^{(i)} \in \mathbb{X}$ to its absolute representation $e_{x^{(i)}} = E_\theta(x^{(i)})$. To transform $e_{x^{(i)}}$ to relative representation, a subset $\mathbb{A} \subset \mathbb{X}$ is chosen as the anchor set. For every training data $x^{(i)}$, a cosine similarity score

$$S_C(e_{x^{(i)}}, e_{a^{(j)}}) = \frac{e_{x^{(i)}} e_{a^{(j)}}}{||e_{x^{(i)}}|| ||e_{a^{(j)}}||}$$

is calculated with respect to $a^{(j)} \in \mathbb{A}$. Then, the relative representation is calculated as

$$r_{x^{(i)}} = (S_C(e_{x^{(i)}}, e_{a^{(1)}}), \ldots, S_C(e_{x^{(i)}}, e_{a^{(|\mathbb{A}|)}}))$$

To generate the anchor word set, we did a single random sample of 200 English words from a uniform distribution over all possible words in our dictionary, following the procedure detailed in Moschella et al. We used the Mandarin translation equivalents of the English words to form the Mandarin anchor word set.

2.3 Model Design

To translate from English to Mandarin, we developed an En-Zh encoder-decoder model, trained on our custom dictionary. The model uses 200-dimensional relative embeddings for English input. During training, the encoder compresses the information from these embeddings into a latent space. This encoded information is then mapped to its corresponding Mandarin translations by a pre-trained decoder. The decoder utilizes weights from a Zh-Zh autoencoder trained specifically for this mapping process, enabling effective translation from English to Mandarin. The code can be found here[2].

2.3.1 Zh-Zh Autoencoder

The Zh-Zh autoencoder was trained to learn the weights connecting the Chinese relative embedding layer to its one-hot vector representation (a binary vector where only one element is 1, indicating the presence of the Chinese word, and all other elements are 0). As shown in the left of Figure 1, the relative embedding for a specific Chinese word is selected by the one-hot vector. The autoencoder then learns the weights that transform the embedding back to the corresponding one-hot vector. To expedite training, we initialized the weights for this mapping as the transpose of the pretrained weights from the one-hot vector to the embedding

[2] https://github.com/Jenniebn/wordLevelTrans

layer rather than random initialization. The learned weights were then used in the decoder of the En-Zh model to map the Chinese embeddings back to one-hot vectors. The autoencoder was trained using the Adam optimizer with cross-entropy loss, with a starting learning rate of 0.01.

2.3.2 En-Zh Encoder Decoder

Given the possibility of multiple correct Mandarin translations for each English word, the En-Zh model's training objective is framed as a multi-label classification task. The model aims to predict a set of Mandarin translations by learning the mapping between the English and Mandarin latent spaces. As shown on the right of Figure 1, a random set of one-hot encoded English words are input to the model, and processed through a 75-dimensional hidden layer with leaky ReLU activation. With frozen weights from the Zh-Zh autoencoder, the decoder converts the vector into corresponding vectors representing the translated Mandarin words. A trainable bias term is added before the output to adjust the decision threshold from 0.5. A binary cross-entropy (BCE) loss weighted by positive classes is employed to address the class imbalance. The model is trained using the Adam optimizer with an initial learning rate of 0.01.

The positive class weight for the BCE loss was determined empirically. Initially, without a positive class weight, the model failed to predict any translations, as the penalty for incorrect predictions was too small. Given that only a few out of 95,685 possible Mandarin words corresponded to the correct translations, the model defaulted to predicting zero for every Chinese word, effectively avoiding any meaningful output. Conversely, when following the recommended positive class weight from the documentation (PyTorch, 2025)—where the weight is set based on the ratio of negative to positive examples—the model produced excessively high recall, generating a wide range of Mandarin words with little precision. After empirical tuning, it was found that using just 2% of the recommended positive weight provided the best balance, significantly improving precision while controlling recall.

3 Results

3.1 Model Performance

Given the challenge of selecting the correct Mandarin translations from nearly 100,000 possible words, our primary focus is not on achieving high absolute performance but rather on analyzing the model's relative performance across different word categories. Despite this inherent difficulty, after training, the model achieved an F1 score of 0.004 on the test set, which is 40% of its training F1 score (0.01), as shown in Table 1. This suggests that the model generalizes its learned patterns to new data, even if overall performance remains low. Notably, the model favors recall over precision, capturing many possible Mandarin translations for each English word but often failing to match the exact dictionary translations.

Table 1: Model performance in training, validation and testing dataset

	Macro Metric		
	Precision	Recall	F1
Training	0.006	0.035	0.01
Validation	0.003	0.006	0.004
Testing	0.003	0.006	0.004

3.2 Word Class Performance

Our model performs better on concrete words and emotional words as shown by Table 2, with a significant difference in the translation accuracy of concrete vs. abstract ($p < 0.001$), concrete vs. unknown ($p < 0.001$), and emotional vs. non-emotional ($p < 0.005$), indicating that translation accuracy is driven by both the concreteness and emotionality of a word. Out of all classes, the best performance is achieved on the concrete emotional words with a translation accuracy of 14.36% on the testing set.

We hypothesized that the model would translate concrete words with the highest accuracy as they represent tangible, physical objects. For example, a table is the same in America and China, but the feeling of shame in English may have different cultural or linguistic subtleties in Chinese. As shown by Table 2, out of all word classes, the model translates the concrete words with higher accuracy than the other 2 classes. Similarly, we hypothesized that emotional words would be more accurately translated than non-emotional words as they represent concepts that are highlighted and more richly defined by their emotional properties, and thus more narrow in the contexts in which they can be used.

Table 2: Model Performance on Word Classes in the Testing Set

Word Class	Emotion Class	Size	Translation Accuracy	Example
Concrete	Emotional	195	**14.36%**	grave, sweet
	Non-Emotional	684	8.48%	scallion, raincoat
Abstract	Emotional	299	5.69%	improve, depressed
	Non-Emotional	536	4.66%	control, overall
Unknown Abstraction	Emotional	42	4.76%	committed, bothering
	Non-Emotional	914	3.39%	biking, roadbed

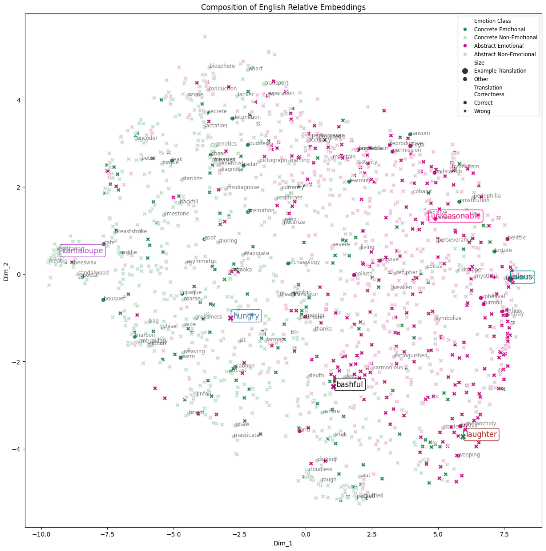


Figure 2: English Embedding Space from the Testing Data

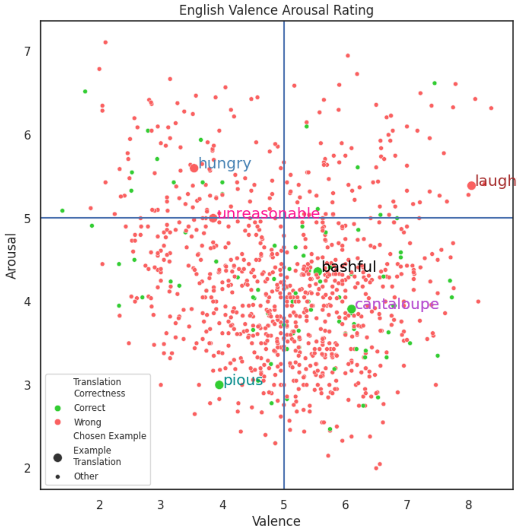


Figure 3: English Valence and Arousal Ratings from the Testing Data

3.3 Error Analysis

In order to better investigate how emotionality and contextual similarity are preserved between languages, we undertook a qualitative error analysis comparing the distribution of English input words to the distribution of model outputs in the Mandarin embedding and valence/arousal spaces. We broke down the different types of words that the model errs on into three dimensions of analysis. For the purpose of this analysis, we only looked at words with multiple outputs and valence and arousal ratings in both languages.

First, we observe whether the model outputs for each word are spread out or if they cluster in a particular area. We also check the distance of each cluster of outputs for a given input word relative to other input words and their clusters. As part of this, we examine how similar the distances between input words in English embedding/valence spaces are to distances between output clusters in the Mandarin embedding/valence spaces. Lastly, we see whether the valence and arousal of input words in the English spaces are similar/in the same areas as their output clusters and target Mandarin equivalents. By looking at which words our model exhibits with what combinations of behavior, we can infer the different types of error and why they may have occurred.

The first type of error occurs with input words that have the following two features. One, their outputs group together in the Chinese embedding space in similar ways to words near them in the English embedding space. Two, they have similar valence/arousal to the various Chinese outputs. One example is the word "cantaloupe", seen in Figures 2, 3, 4. When the model errs on a word in this way, it fails to return one of our expected target translations, but it often still has outputs that group together near where our target term is in the embedding space. Errors on words like these show our model is good at finding regions of the semantic space that contain words similar to a target rather than narrowing in on the specific word itself. These errors are expected, as in these cases our model learns an appropriate approximate mapping between the lexical semantic spaces, but this mapping does not contain the best translation(s) given in dictionaries. As we obtained a set of correct translations for the model to reference via dictionary validation rather than human rating or parallel corpora, our “correct” translation set is somewhat inflexible and potentially not entirely representative of possible translations defined by real language use.

The second type of error appears with words that have model outputs that are spread out in both the Mandarin embedding and valence/arousal spaces, such as "hungry". Our model erring on such words implies an issue with either our data or our model architecture/parameters, such that our model cannot make confident guesses on what such words look like when translated.

The third type of error involves clustering and a similar structure between spaces as in the first type of error, but it also shows specific discrepancies in emotionality such as flipped valence or arousal in the Mandarin valence/arousal space. Such examples appear to have model outputs with strong clustering, and investigation into output meanings shows the potential for such errors to be due to cross-cultural differences in the given words. In "bashful", for example, outputs hone in around a higher arousal value as opposed to its negative arousal value in English, and the outputs are words like "sexy". These discrepancies hint at these specific words being conceptualized differently in Chinese but still having solid enough associations for our model to have confident guesses about them, albeit being incorrect, possibly as a result of these words being more difficult to translate between these languages for specific cultural differences.

4 Discussion

4.1 Implications/applications of Results

In this paper we have proposed a computational method of exploring how transferable the dimensions of emotion and abstraction are cross-linguistically. We hypothesized that a word level machine translation model could learn how to align the semantic spaces of two given languages, which would then provide a direct method of investigating how words are retrieved across languages along these dimensions of emotion and abstraction.

As hypothesized, our model had better translation performance for concrete and emotional words than for other words, mirroring the patterns of human participant results. We specifically compared our results to "simultaneous bilinguals", as finding participant groups with nearly equal native-level fluency in two languages theoretically controls for language proficiency. (Ferré et al., 2017).

Congruent to previous psycho-linguistic literature, our model has higher accuracy on concrete/weak concrete words as opposed to abstract words (Guasch and Ferré, 2021; Ferré et al., 2017). Intuitively, this makes sense, as concrete words have more imageable referents in the world compared to more abstract concepts. While our model has no built-in cognition of referents in the world, it can learn patterns of contextual usage that may differentiate concrete words from abstract ones. Furthermore, when data was sufficient, our model showed higher accuracy on translations of emotion-laden/label words than on unknown/non-emotional words. This also agrees with prior literature (Kousta et al., 2011; Ferré et al., 2017).

This human-model congruence provides further evidence for the presence of certain distinct features that make "emotional" and "concrete" words more recognizable than their neutral and abstract counterparts, respectively. Previous literature has investigated the effect of emotionality and abstraction within languages of simultaneous bilinguals (Ferré et al., 2017).

Our model uses pre-trained word embeddings, which are developed from the contexts in which given words are used. Given this, our model better recognizing concrete and emotional words could mean that these word types have greater consistency in their contexts compared to their abstract and non-emotional counterparts. Similarly, increased concreteness and abstraction of words have been shown also to facilitate word processing in human participants. This suggests that context can be utilized to detect words that represent concepts that are more recognizable/processable due to such values. More direct confirmation of the encoding of concreteness and abstraction in context and embeddings could be checked for via performance analysis of a concreteness/emotionality classifier’s agreeability with human ratings.

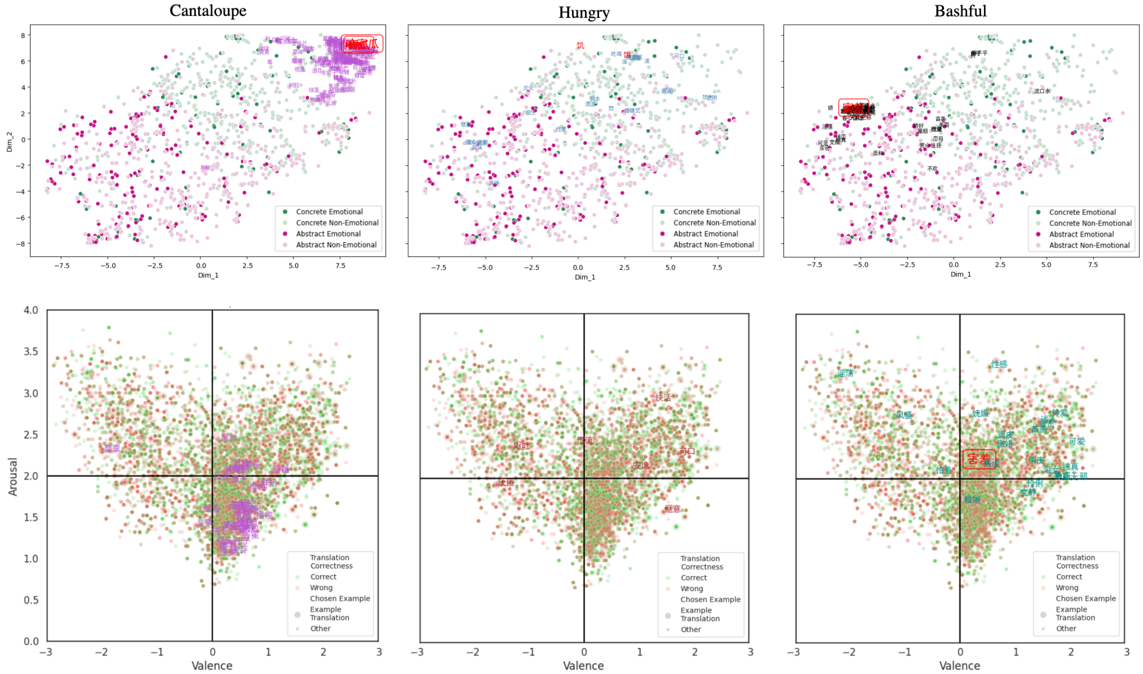


Figure 4: Mandarin Embedding Space Examples

One avenue of further research would be to validate our findings with English and Mandarin simultaneous bilinguals. As previous investigations into emotion and abstraction have often used within-language tasks (Ferré et al., 2017), an interlingual lexical decision task that presents both Mandarin and English stimuli within one experiment could provide more insight into how emotion and abstraction are processed in cross-linguistic contexts.

The agreement of the model with human trends of emotion/abstraction processing suggests potential for further research into the utilization of word-level models as a point of comparison to human processing of similar affect categories as explored here. These models could be used as tools to assist with experiments that would typically require hard-to-recruit participant groups, specifically simultaneous bilinguals. As our model requires pre-trained monolingual embeddings from two languages, rather than parallel translation data, it could be more accessible than recruiting simultaneous bilinguals for preliminary investigation depending on the language groups one wishes to study.

To extend more directly on this study, one could investigate other languages in addition to Mandarin and English in a similar model architecture as ours to see if results vary as a function of language relatedness. One potential option could be Japanese, to distinguish the effects of historical influence and linguistic relatedness. This could be a new way to investigate how universal the concepts of emotionality/concreteness are in human cognition.

4.2 Error Analysis Implications and Applications

Looking back at the error analysis in Section 3.3, a question arises as to what implications/applications we can discern from the three kinds of errors described earlier. Recall that one of the dimensions of error is whether the model outputs are located in the same approximate region of the lexical and emotional space as their input. Depending on how similar/dissimilar inputs and outputs are on this metric, different errors can be considered “more correct” or “less correct” than others.

This has interesting implications in the context of the third type of error, which involves words like "bashful", i.e., those that retain strong output clustering and similarity between embedding spaces, but vary in valence and/or arousal across the spaces. Many words of the third error type also have Mandarin outputs that intuitively seem more semantically dissimilar to the English input than expected. One such example is our model relating “bashful” to Chinese outputs that comment on attractiveness, like “sexy”. This suggests that the acceptable contexts in which to use a word vary as a function of society/culture. This also aligns with recent semantic association research which found that cross-linguistic semantic alignment of sets of concepts is heavily impacted by the levels of cultural similarity between the speakers of given language pairs. (Thompson et al., 2020). Further investigation is warranted to quantify how cross-cultural variation may interfere with or facilitate the mapping of concepts across languages, and how to better contextualize cross-linguistic research results by it.

The arousal/valence of both target words and their associated output clusters differing across languages in such cases implies that some concepts, and the contexts their representations are used in, can vary exceptionally depending on cross-cultural differences. This suggests promising applications for using further statistical/machine learning models to quantify how emotional sentiment can vary cross-culturally within and across languages as a factor of various cultural categories, such as religion or types of personality traits. Furthermore, a question arises as to whether or not congruence of cross-linguistic emotional sentiment is a confounding variable in machine translation model performance.

5 Conclusion

This research developed a neural network model using relative word embeddings to investigate the impacts of emotionality and abstraction on a bilingual semantic space mapping. Our model's maximum accuracies were 14.36% for concrete emotional words and 8.48% for concrete non-emotional words. An in-depth error analysis revealed that although the model didn't learn word-to-word mapping, it generally achieved a mapping of subregions onto each other, with a handful of errors being due to a lack of data and cultural differences impacting word representations. The model's performance agrees with previous results of emotional and concrete words providing a processing advantage, and furthermore suggests that this processing advantage is cross-lingual.

Limitations & Future Work

Our most glaring limitation is the issue of polysemy - a word having multiple meanings. Polysemy can lead to lower translation accuracy due to differing levels of emotionality and abstraction in the different meanings of polysemous words such as "grave". Some secondary limitations are that our embedding visualization compresses a 200 dimensional semantic space into 2 dimensions, leading to information loss, and that we use full correctness as a criterion for the model. Utilizing an information theoretic measure such as cross entropy would allow for more flexibility and sensitivity, and could reduce the impact of polysemy as well. Finally, our model with one hidden layer restricts the amount of complex information it can learn. For further research we suggest taking polysemy into greater consideration and increasing the complexity of the neural network model. Another interesting extension of our work would be validating our results with an English-Mandarin simultaneous bilingual population, which would provide a direct comparison of human vs. machine performance and serve as a benchmark for future emotionality or simultaneous bilingual research.

References

Jeanette Altarriba and Lisa M. Bauer. 2004. The Distinctiveness of Emotion Concepts: A Comparison between Emotion, Abstract, and Concrete Words. *The American Journal of Psychology*, 117(3):389.

Jeanette Altarriba, Lisa M. Bauer, and Claudia Benvenuto. 1999. Concreteness, context availability, and imageability ratings and word associations for abstract, concrete, and emotion words. *Behavior Research Methods, Instruments, & Computers*, 31(4):578–602.

Jeanette Altarriba and Tina Sutton. 2004. The influence of emotional arousal on affective priming in monolingual and bilingual speakers. *Journal of Multilingual and Multicultural Development - J MULTILING MULTICULT DEVELOP*, 25:248–265.

Gerry T. M. Altmann. 2001. The language machine: Psycholinguistics in review. *British Journal of Psychology*, 92(1):129–170.

J. Binder, Chris Westbury, K. McKiernan, E. Possing, and D. Medler. 2005. Distinct brain systems for processing concrete and abstract concepts. *Journal of Cognitive Neuroscience*, 17:905–917.

Steven Bird, Edward Loper, and Ewan Klein. 2009. Natural language processing with python.

Katarzyna Bromberek-Dyzman, Rafał Jończyk, Monica Vasileanu, Anabella-Gloria Niculescu-Gorpin, and Halszka Bąk. 2021. Cross-linguistic differences affect emotion and emotion-laden word processing: Evidence from Polish-English and Romanian-English bilinguals. *International Journal of Bilingualism*, 25(5):1161–1182.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior Research Methods*, 46(3):904–911.

Cambridge. 2024. Cambridge English–Chinese (Simplified) Dictionary: English to Mandarin Chinese.

CC-CEDICT. CC-CEDICT Home [CC-CEDICT WIKI].

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Pilar Ferré, Manuel Anglada-Tort, and Marc Guasch. 2017. Processing of emotional words in bilinguals: Testing the effects of word concreteness, task type and language status. *Second Language Research*, 34(3):371–394.

Ralph Grishman. 1989. *Computational linguistics: An introduction*. Cambridge UP.

Marc Guasch and Pilar Ferré. 2021. Emotion and concreteness effects when learning novel concepts in the native language. *Psicológica Journal*, 42(2):177–191.

J. A. Hinojosa, E. M. Moreno, and P. Ferré. 2020. Affective neurolinguistics: towards a framework for reconciling language and emotion. *Language, Cognition and Neuroscience*, 35(7):813–839.

Daniel Jurafsky and James Martin. 2008. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, volume 2.

Stavroula-Thaleia Kousta, Gabriella Vigliocco, David P. Vinson, Mark Andrews, and Elena Del Campo. 2011. The representation of abstract words: Why emotion matters. *Journal of Experimental Psychology: General*, 140(1):14–34.

Emiel Krahmer. 2010. What Computational Linguists Can Learn from Psychologists (and Vice Versa). *Computational Linguistics*, 36(2):285–294.

Xiaogen Liao and Chuanbin Ni. 2022. The effects of emotionality and lexical category on L2 word processing in different tasks: Evidence from late Chinese–English bilinguals. *Quarterly Journal of Experimental Psychology*, 75(5):907–923.

Wei Lin. 2024. skywind3000/ECDICT. Original-date: 2017-03-20T15:03:10Z.

Asifa Majid. 2012. Current Emotion Research in the Language Sciences. *Emotion Review*, 4(4):432–443.

MDBG. MDBG English to Chinese dictionary.

Filiz Mergen and Gulmira Kuruoglu. 2017. A Comparison of Turkish-English Bilinguals' Processing of Emotion Words in Their Two Languages. *Eurasian Journal of Applied Linguistics*, 3(2):89–98.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. 2022. Relative representations enable zero-shot latent space communication. *ArXiv*, abs/2209.15430.

Sophie Pauligk, Sonja Kotz, and Philipp Kanske. 2019. Differential impact of emotion on semantic processing of abstract and concrete words: Erp and fmri evidence. *Scientific Reports*, 9:1–13.

Aneta Pavlenko. 2012. Affective processing in bilingual speakers: Disembodied cognition? *International Journal of Psychology*, 47(6):405–428.

Marta Ponari, Sara Rodríguez-Cuadrado, David Vinson, Neil Fox, Albert Costa, and Gabriella Vigliocco. 2015. Processing advantage for emotional words in bilingual speakers. *Emotion*, 15(5):644–652.

PyTorch. 2025. Bcewithlogitsloss - pytorch 2.3 documentation.

Peter H Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Humera Sharif and Saqib Mahmood. 2023. Emotional processing in bilinguals: A systematic review aimed at identifying future trends in neurolinguistics. *Humanities and Social Sciences Communications*, 10(1).

Bill Thompson, Seán Roberts, and Gary Lupyan. 2020. Cultural influences on word meanings revealed through large-scale semantic alignment. *Nature Human Behaviour*, 4:1–10.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Liwei Wang, Lunjia Hu, Jiayuan Gu, Zhiqiang Hu, Yue Wu, Kun He, and John Hopcroft. 2018. Towards understanding learning representations: To what extent do different neural networks learn the same representation. *Advances in neural information processing systems*, 31.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behavior rResearch mMethods*, 45(4):1191–1207.

Yabla. Chinese English Dictionary with Pinyin, Strokes, & Audio - Yabla Chinese.

Barbra Zupan, Lynn Dempsey, and Katelyn Hartwell. 2023. Categorising emotion words: the influence of response options. *Language and Cognition*, 15(1):29–52.

A Theory of When and How Learners Construct Tiers: Implications for Opaque and Transparent Vowels

Caleb Belth
University of Utah
caleb.belth@utah.edu

Abstract

Some vowel harmony systems have neutral vowels, which need not agree along the harmonizing dimensions of vowel quality. Neutral vowels differ in whether other vowels in turn harmonize with them: those that are harmonized with are *opaque* while those that are not are *transparent*. Prior artificial language learning studies have found opaque vowels to be more readily learned in laboratory settings than transparent vowels. This was initially thought to be because transparent vowels intervene between harmonizing vowels on a vowel tier, making harmony non-local. However, subsequent computational work has demonstrated that vowel harmony is typically tier-strictly-local, even with transparent and opaque vowels, indicating that there may be less difference between them than once believed. I propose an explanation for the different learning results between transparent and opaque vowels by making use of a recent learning model that proposes learners create tier-like representations in response to being unable to sufficiently generalize without them, as measured by the Tolerance Principle. I demonstrate how the representations that this model constructs make sense of different learning results between transparent and opaque vowels, despite their shared formal properties.

1 Introduction and Background

Vowel harmony involves non-local dependencies, as vowels agree along the harmonizing dimensions across intervening consonants. In the following example (1) from Turkish, the underlined suffix vowels harmonize in backness with the vowel to their left (Nevins 2010, p. 28; Kabak 2011, p. 3).

(1)	[dɑl-lɑr-ɯn]	branch-PL-GEN
	[jer-ler-in]	place-PL-GEN
	[ip-ler-in]	rope-PL-GEN

In some vowel harmony systems, a subset of vowels are not required to harmonize—they are *neutral*. These neutral vowels are coarsely grouped into two categories: *opaque* and *transparent*. Opaque vowels participate in harmony in that other vowels harmonize with them. For example, in addition to backness harmony, Turkish high vowels [i, y, ɯ, u] also harmonize in roundness (2a). Low vowels [e, ø, ɑ, o] are neutral to the rounding harmony (2b), but high vowels nevertheless harmonize with them *opaquely* (2c).

(2)	a.	[ip-in]	rope-GEN
		[jyz-yn]	face-GEN
		[kɯz-ɯn]	girl-GEN
		[buz-un]	ice-GEN
	b.	[kɯz-lɑr]	gril-PL
		[buz-lɑr]	ice-PL
	c.	[el-in]	hand-GEN
		[søz-yn]	word-GEN
		[sɑp-ɯn]	stalk-GEN
		[jol-un]	road-GEN

Transparent vowels, on the other hand, are inert, neither harmonizing nor being harmonized with. For instance, while Hungarian has backness harmony (3a), the vowels [iː, eː] are transparent, with the DAT vowel skipping them to harmonize with the next vowel to the left of (3b; examples from Benus and Gafos 2007).

(3)	a.	[ørøm-nɛk]	joy-DAT
		[moːkuʃ-nɔk]	squirrel-DAT
	b.	[ɛmiːr-nɛk]	emir-DAT
		[pɔpiːr-nɔk]	paper-DAT
		[myːveːs-nɛk]	artist-DAT
		[kaːveː-nɔk]	coffee-DAT

The development of autosegmental theory (Goldsmith, 1976) allowed for treating vowel harmony as local on a vowel tier (Clements, 1976, 1980). Opaque vowels do not harmonize with the preceding vowel on a vowel tier, but their features take over the harmony, so all remains local. However, vowel harmony must cross transparent vowels, which introduces non-locality even on a vowel tier (Goldsmith, 1985; Bakovic and Wilson, 2000; Hayes and Londe, 2006; Finley, 2009). Finley (2015) hypothesized that this makes transparent vowels harder to learn than opaque vowels and tested this hypothesis with a series of artificial grammar learning (AGL) experiments. Finley found that adults indeed succeeded at learning the behavior of an opaque vowel but failed to learn the behavior of a transparent vowel under equivalent conditions. Only by increasing the amount of evidence of the neutral vowel's transparency, by increasing the amount of exposure to items that unambiguously indicated transparency, did learners eventually succeed at learning transparent vowel behavior. Chen (2024) found compatible results: when adults were trained on an artificial harmony system with a neutral and a transparent vowel, they either failed to learn the harmony system altogether or appeared to treat both the opaque and transparent vowels as opaque (depending on the presentation of the training stimuli).

However, work in computational phonology has found that from a formal-language-theoretic perspective, neither opaque nor transparent vowels meaningfully change the computational character of vowel harmony: vowel harmony is typically tier-strictly-local ($k = 2$) (Heinz et al., 2011), with or without opaque and/or transparent vowels (Burness et al., 2021). Learners could project a tier that excludes transparent vowels along with the consonants, and this renders all relevant dependencies local on the tier. Moreover, as Finley (2015) observed, transparent vowels must be learnable, since they appear in numerous natural language harmony systems. Indeed tier-strictly-local constraints and processes are provably efficiently learnable (Jardine and Heinz, 2016; Jardine and McMullin, 2017; Burness and McMullin, 2019) and Finley (2015) did find that under the right conditions, transparent vowel harmony can be learned in the lab. Similarly, Ozburn et al. (2016) found that adult Canadian French speakers succeeded at learning the behavior of a transparent vowel in an artificial vowel harmony system built around the French vowel inventory.

Given that vowel harmony with opaque and transparent vowels shares a fundamental underlying computational structure and both must be learnable in natural languages, it is worth revisiting what might underlie the picture from experimental results that vowel harmony is harder to learn with transparent vowels than opaque vowels.

To do so, I build on my prior work (Belth, 2024), where I proposed that humans learn phonological alternations by tracking dependencies between alternating segments and the segments adjacent to them—using the well-attested ability to track adjacent dependencies over many kinds of representations (Saffran et al., 1996, 1997; Aslin et al., 1998; Saffran et al., 1999; Fiser and Aslin, 2002). In that proposal, if adjacent dependencies are not sufficiently predictive of the alternation, where *sufficiency* is measured by the Tolerance/Sufficiency Principle (Yang, 2016), learners use the same sensitivity to adjacent dependencies to form a new representation that excludes any adjacent segments that led to incorrect predictions. The resulting representations can be interpreted as tiers, which are constructed in dynamic response to the input. In Belth (2024), I implemented this proposal as a learning model. The model succeeded at learning natural language harmony processes, including Turkish vowel harmony, in which low vowels are opaque to rounding harmony, and Finnish vowel harmony, in which, similarly to Hungarian, [i, e] are transparent to backness harmony (Ringen and Heinämäki, 1999). In Turkish, the learner constructed a vowel tier and in Finnish it constructed a tier that excluded the transparent vowels. Thus, the proposal already accounts for the learnability of vowel harmony with opaque and transparent vowels in natural languages. In this paper, I will demonstrate that it simultaneously accounts for the difference in experimental settings between artificial vowel harmony systems with opaque vs. transparent vowels.

Consider a transparent vowel harmony system, such as the artificial one from Finley (2015), where a suffix vowel harmonizes in backness with the final vowel of the stem (4a), but where the vowel [ɛ] is neutral (4b)-(4c). Since the neutral vowel is itself front, only when the penultimate stem vowel is back (4b) do we get unambiguous evidence that [ɛ] is transparent.

(4) a. [budok-o]
[degib-e]
b. [dotɛb-o]
c. [tedɛt-e]

It is thus possible that the learner treats the neutral vowel as *opaque* and handles the cases like (4b), which contradict this, as lexicalized exceptions. If, during learning, enough of these exceptions accumulate that the learner's harmony generalization is no longer tenable with them as exceptions (which, as in Belth 2024, will be measured with the Tolerance principle), then the learner will again change representations, excluding the neutral vowel because it is no longer sufficiently predictive, thereby rendering it transparent. Thus, transparent vowels can for a time be tolerated as opaque vowels with lexicalized exceptions. This is the main idea underlying my proposed explanation for the observed experimental differences in learning.

In the next section § 2, I introduce the model from Belth (2024) (D2L) in more detail. In § 3, I survey prior experimental work on learning transparent and opaque vowels. I then demonstrate how D2L accounts for these experimental results, as conceptually described above, and also demonstrate that a number of other models fail to account for them § 4. I conclude with a discussion § 5.

2 Model

The model from Belth (2024), named D2L, was based on the developmental trajectory of children's ability to track adjacent and non-adjacent dependencies. Children show evidence of tracking adjacent dependencies at a younger age—as young as 8 months (Saffran et al., 1996, 1997; Aslin et al., 1998)—than tracking non-adjacent dependencies, which appears to develop around 15-18 months (Santelmann and Jusczyk, 1998; Gómez, 2002; Gómez and Maye, 2005). Tracking of adjacent dependencies has been observed over a range of different kinds of structures, linguistic and non-linguistic, including shapes (Fiser and Aslin, 2002) and non-linguistic tones (Saffran et al., 1999). These results serve as evidence of a language-independent psychological mechanism—the ability to track adjacent dependencies—that could underlie the learning of phonological alternations.

D2L implements the proposal that when learning a phonological alternation, a learner's attention is drawn to the alternating segment, and they begin tracking segments adjacent to it. I will use Finley (2015)'s artificial vowel harmony system as an example (see § 3) to describe the model as it pertains to the present paper. In (5), the underlying /-V/ suffix alternates between [-e] ~ [-o].[1]

(5) /budok-V/ → [budoko]
/degib-V/ → [degibe]
/gemit-V/ → [gemite]
/kukop-V/ → [kukopo]
/tedɛt-V/ → [tedɛte]
/dotɛb-V/ → [dotɛbe]

D2L's attention is centered around /-V/ and the segments adjacent to it—here, the stem-final segments. D2L attempts to enforce harmony using the final segments, but since they are all consonants, the harmony fails. The learner then creates a new representation, excluding any adjacent segments that harmonizing with fails to yield the observed surface form for /V/—here /k, b, t, p/. D2L attempts to form a natural class for these segments, in this case [−syl]. The new representation is the complement of this *deletion set*, namely [+syl]. Clearly, this has the interpretation of a vowel tier (6).

(6) /uo-V/ → [uoo]
/ei-V/ → [eie]
/ei-V/ → [eie]
/uo-V/ → [uoo]
/eɛ-V/ → [eɛe]
/oɛ-V/ → [oɛe]

D2L then tracks segments adjacent to /-V/ on this new representation. The vowel [ɛ] here is opaque, so harmonizing with the adjacent vowel on this representation yields the expected surface realizations of /-V/ and D2L has succeeded in forming a representation and generalization that sufficiently accounts for the alternation. Following the notation from Belth (2024), (7) shows the generalization, where the vowel /V/ agrees in the value for feature [back] with an adjacent [+syl] segment after projecting vowels.

(7) AGREE(V, [back]) / [+syl] __ ∘ proj([+syl])

If, on the other hand, the vowel [ɛ] were transparent, the surface form of /dotɛb-V/ would be [dotɛbo], in which case enforcing harmony on the new representation would yield the wrong surface form for /-V/: *[e] instead of [o] (8).

(8) /oɛ-V/ → *[oɛe]

In this way, stems where a back vowel precedes a transparent front vowel will be exceptions to the generalization D2L forms on the new representation. D2L changes representations whenever the

[1] See Belth (2023a,b) for a proposal on how learners might come to attend to learning an alternation in the first place, and where the underlying forms might come from.

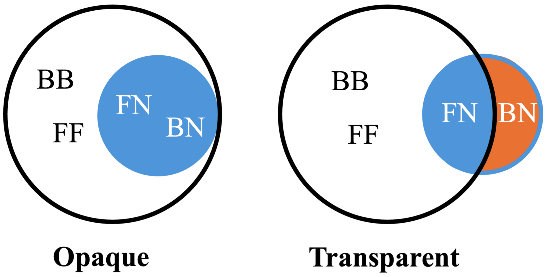


Figure 1: B = Back, F = Front, N = Neutral (opaque or transparent, depending on condition). The black circle represents the stems that are predictable from an adjacent vowel once D2L has constructed a vowel tier. The blue circle represents stems where the adjacent vowel is neutral. The orange sub-circle represents the only stems for which the suffix is not predictable from the tier-adjacent vowel.

generalization it forms over its current representation fails to sufficiently account for the alternation. D2L uses the Tolerance Principle (TP; Yang 2016), which has been evaluated in experimental settings (Schuler et al., 2016; Shi and Emond, 2023), to decide whether the generalization can sustain a particular number of exceptions (9).

(9) **Tolerance Principle**: a rule applying to n items with e exceptions is productive *iff* $e \leq \frac{n}{\ln n}$

Thus, D2L will only change representations again if the number of exceptions due to harmonizing with the transparent vowel, relative to the total number of alternating items, rises above the TP threshold (9). If the number of exceptions fall below the threshold, then D2L lexicalizes the exceptions and may overextend harmony with the final vowel (7) to new words with a final transparent vowel. On the other hand, if the number of exceptions grows too large, D2L will recursively construct a new representation, this time excluding the vowel [ɛ]—the culprit behind the exceptions—in addition to the consonants, as (10) shows.

(10) Agree(V, [back]) / [+syl] ___
∘ proj([+syl] \ {ɛ})

This core idea is visualized in Figure 1. Once D2L has constructed a new representation that excludes consonants (i.e., a vowel tier), the suffix vowel is entirely predictable from the newly-adjacent vowel if the neutral vowel (N) is opaque. This set of stems, for which the suffix is adjacently predictable, is represented by the large black circle.

Table 1: The four basic kinds of training items in Finley (2015)'s study. B = Back, F = Front, N = Neutral (opaque or transparent, depending on condition). The right two columns give the suffix corresponding to the condition (only the BN items differ between conditions)

Kind	Types	Example	Opaque	Transparent
BB	8	[budok]	[-o]	[-o]
FF	8	[degib]	[-e]	[-e]
FN	4	[tedɛt]	[-e]	[-e]
BN	4	[dotɛb]	[-e]	[-o]

On the other hand, if the neutral vowel is transparent, only BN words are not adjacently predictable (the orange sub-circle). But if the orange part of the diagram is small enough, then it may be relegated to lexicalization, at least for a time.

3 Prior Experimental Studies

Finley (2015) carried out a series of artificial grammar learning studies with adults, involving opaque and transparent vowels. Finley first compared each of two experimental groups—one Opaque and one Transparent—to relevant control groups. The experimental groups were trained on CVCVC nonce words, each of which could be suffixed with either front [-e] or back [-o]. The artificial language also had the vowels [i, u] and the neutral vowel [ɛ], which only occurred as the final vowel. The choice of suffix was based on harmony with the final stem vowel, except for the words in the Transparent condition that had the transparent [ɛ] as the final vowel; for these the choice was based instead on harmony with the penultimate vowel. This is summarized in Table 1. There were 8 stems each with two harmonizing vowels (8 BB and 8 FF), 4 stems with a front vowel before the neutral [ɛ] (FN), and 4 with a back vowel before it (BN).

In the Opaque condition, if learners choose the suffix based on the final stem vowel, they would accurately generalize to test words of all four kinds. In the Transparent condition, however, accurate generalization to test BN words would require learning the transparency of [ɛ]. In other words, because [ɛ] is front, only BN words show unambiguous evidence that [ɛ] is transparent rather than opaque. Finley's first experiment, which presented each stem-suffixed pair 5 times, suggested that the participants in the Opaque condition learned vowel harmony, including the behavior of the opaque vowel. However, participants in the Transparent condition learned the basic vowel harmony pattern, but showed no evidence of learning the behavior of the transparent vowel.

Finley then attempted to find conditions in which participants would succeed at learning the transparent vowel's behavior. In a second experiment, the 4 FN words, for which it is ambiguous whether the [-e] vowel is harmonizing with the final or penultimate vowel (which are both front), were replaced with 4 additional BN words (all taking [-o]). This decreased the learners' test performance across the board. One interpretation is that because the suffix [-o] became more dominant—now occurring with 2/3 of training items—learners failed to attend to learning the alternation at all.

Finley then returned to the original setup (balanced items between FN and BN), and tried replacing the neutral vowel [ɛ] with [ɪ]. The participants again learned the overall harmony pattern, but not the transparent vowel. In another experiment, each word was presented 10 times instead of 5. This led to an increase in performance on the transparent vowel, but the increase over the control group was not statistically significant. The next experiment added 6 additional unambiguously transparent (BN) stems, with all words being presented 10 times. This also led to an increase, though not statistically significant, in performance on the transparent vowel. Finally, increasing the number of presentations of the BN stems to 20, while keeping the others at 10, led to an increase in performance on the transparent vowel that was significantly higher than the control group's.

The overall picture is that under some conditions where adults will learn a vowel harmony system with an opaque vowel, they will fail to learn a transparent vowel. But, if sufficient exposure to words that demonstrate the transparency of a vowel is available, adults will succeed at learning its transparency. While this overall picture is clear, the precise conditions in which learning a transparent vowel will or will not succeed are less so. In multiple of Finley (2015)'s experiments, the results showed a numerical increase in performance that was not statistically significant. The number of participants in some experiments was small (often $<$ 20 per condition), thus warranting a level of caution in drawing strong conclusions from any particular significance test. The study involved adults, but we also know that children acquire vowel harmony systems with transparent vowels (MacWhinney, 1978; Gósy, 1989; Leiwo et al., 2006; Gonzalez-Gomez et al., 2019). Moreover, the stimuli were presented in auditory form only, with no accompanying image. It is difficult to know in such a scenario whether participants treated multiple tokens of the same type as in fact being part of a single word type or of multiple. Consequently, the relative role of type and token frequencies is not entirely clear.

Furthermore, Ozburn et al. (2016) note that Finley's artificial language used the English vowel inventory, which leads to both roundness and backness alternating ([-e] is front, unround; [-o] back, round), which is not typical in natural language backness harmony with transparent vowels. Ozburn et al. trained adult Canadian French speakers in a similar setting as Finley's, but using harmony centered around the French vowel inventory, which includes front rounded vowels, allowing for the rounding dimension to stay fixed. Ozburn et al.'s participants did show evidence of learning vowel harmony transparency in this setting. However, whether this difference in results from Finley's was due to the difference in stimuli and participant populations or to difference in type frequency is not clear: Ozburn et al. do not report how many items of each kind they used in their experiment, but they do say that 1/4 of the items were unambiguously transparent (BN), which is a higher proportion than in Finley's experiments (1/6 to 1/5).

In a related study, Chen (2024) trained adult speakers of Taiwan Mandarin on an artificial vowel harmony pattern with both an opaque and a transparent vowel. The study was primarily interested in a possible "starting small" effect—whether presenting bisyllabic stems before trisyllabic stems, and a disproportionate number of bisyllabic stems, would yield better learning than presenting a balanced number all at once. In the results, only in the "starting small" condition did participants show evidence of learning the vowel harmony pattern. However—more relevant to the current discussion—even in this condition, participants only showed learning of the non-transparent vowels. They appeared to treat the transparent vowel as also opaque. Thus, while this study deviates substantially from the prior two in goals and design, the results largely corroborate the big picture of Finley (2015)'s study: opaque vowels are learned more readily by adults than transparent vowels.

4 Evaluation

To evaluate whether D2L makes sense of the experimental results on opaque and transparent vowels, I tested whether D2L learns an opaque vowel in conditions where it does not learn a transparent vowel (§ 4.2), and whether increasing the amount of training on items showing transparency eventually leads it to learn a transparent vowel (§ 4.3). First, I will introduce the setup (§ 4.1).

4.1 Data and Setup

I used data from Finley (2015)'s study for training and evaluation. As the base training set, I used the same 24 stem-suffixed pairs that Finley, p. 22 reports; these are summarized in Table 1.

In experimental settings (as in natural language learning), participants likely do not learn every word they are trained on. Yet it is over the words that are learned that generalizations can be formed.[2] To simulate this variability in attained vocabulary, I carried out 30 simulations with different samples of training words. For each, I sampled an integer *n* from a Gaussian distribution with mean 20 and standard deviation of 4 to represent the vocabulary size. I then sampled *n* unique words from the 24 training words, weighted by frequency. In the first experiment (§ 4.2) all words were given equal frequency, so the sampling was uniform. In the second experiment (§ 4.3), where the amount of exposure to unambiguously transparent (BN) words is increased, this sampling procedure allows for manipulating the saliency of BN words, as Finley (2015) did, by increasing their relative token frequency.

For testing, I used the novel stems from Finley, p. 23. These include 8 stems with two harmonizing vowels (BB or FF) and 11 ending in the neutral [ɛ]. Of the latter, 9 are BN.

4.1.1 Comparison Models

In a study of vowel harmony in Hungarian, Hayes and Londe (2006) proposed two harmony constraints, applying over a vowel tier. The first, local, constraint incurred a violation whenever a front vowel immediately followed a back vowel on the vowel tier, and the second, distal, constraint incurred a violation whenever a front vowel followed a back vowel anywhere on the tier. The distal constraint was necessary because of Hungarian's transparent vowels. Finley (2015) reasoned that the distal constraint is more complex than the local constraint, and thus could make harmony more difficult to learn when transparent vowels are present. This forms the first comparison model: I trained a Maximum Entropy Harmonic Grammar model using distal and local constraints like Hayes and Londe's. The model learns to map underlying forms (e.g., /dotɛb-V/) to surface forms, using a Maximum Entropy model, as described by Goldwater and Johnson (2003). For each underlying form, two candidates are generated—one with [-o] and one with [-e]—and the number of violations of local and distal harmony constraints are used as the features of each candidate. I will call this model H&L, as an homage to Hayes and Londe (2006).

While H&L learns a Maximum Entropy grammar with provided constraints, it is also possible for constraints to be learned. Indeed, building on Hayes and Wilson (2008)'s model, Gouskova and Gallagher (2020) proposed a Maximum Entropy model that automatically learns to project tiers and form phonotactic constraints over the resulting tier projections. I used the model publicly available from the authors.[3] I will call this model G&G.

Lastly, vowel harmony can typically be characterized as 2-Tier-Strictly-Local (2TSL), whether described as phonotactic constraints (Heinz et al., 2011) or processes (Burness et al., 2021).[4] This is usually true even when opaque or transparent vowels are present. Formal learning algorithms have been proposed that allow for proving the efficient learnability of 2TSL languages and functions (Jardine and Heinz, 2016; Burness and McMullin, 2019). However, while these learnability results apply to vowel harmony with opaque or transparent vowels, it does not necessarily imply that languages with either of these kinds of neutral vowels will be learned at equal rates. Like D2L, the Jardine and Heinz (2016) and Burness and McMullin (2019) models start with a representation where all segments are present, and iteratively remove segments to create new tiers. Unlike D2L, they use the formal properties of TSL to deduce conditions where removing segments is provably correct. Thus, I use TSLIA (Jardine and Heinz, 2016; Jardine and McMullin, 2017), which is publicly available (Aksënova, 2020), as an additional comparison model. Formal models of this family

[2] See, for instance, Schuler (2017, ch. 4) for discussion of this point for artificial language learning with children.

[3] github.com/gouskova/inductive_projection_learner

[4] See Mayer and Major (2018) for an example of a harmony pattern than cannot be characterized as TSL.

often benefit from collapsing pattern-irrelevant differences among segments (Aksënova, 2020; Johnson and De Santo, 2023), which simplifies the learning problem and makes it more likely that the characteristic sample (the information needed in the training data for convergence onto an appropriate grammar) will be present. Following this line of work, I collapsed all consonants into the symbol C, back vowels to B, non-neutral front vowels to F, and neutral vowels to N. This collapsing was only applied to TSLIA's input, not the other models'.

For D2L, I used the implementation publicly available in the Python package *algophon*.[5]

In the experiments, each test stem has two possible suffixed forms: [-e] or [-o]. I compute a model's accuracy based on the fraction of stems for which it produces/chooses the form consistent with the relevant vowel harmony pattern. Specifically, the correct choice for BB and FF is the vowel that agrees in backness with the final stem vowel. In OPAQUE conditions, the correct choice for neutral-vowel-final stems is [-e], while in TRANSPARENT conditions, it is the vowel agreeing with the penultimate stem vowel. I report overall accuracy and neutral-vowel accuracy, which is computed over only the neutral-vowel-final test stems.

This scheme can be interpreted as either learning an alternation (mapping a stem with underlying /-V/ to the surface form) or a phonotactic pattern (learning where [-e] and [-o] can/cannot occur). D2L and H&L learn alternations, while G&G and TSLIA learn phonotactics. At test time, the former are probed to produce a surface form for a stem with the underlying suffix /-V/ and the produced form is taken as the choice. Meanwhile, the phonotactic models are asked to score the two choices and the one with the better well-formedness score is chosen. This setup is identical to Belth (2024)'s.

4.2 Opaque vs. Transparent

The first experiment evaluates whether D2L and the comparison models show a difference in generalization between an OPAQUE vowel harmony condition and a TRANSPARENT condition (learning the former better). The experiment uses the training data described above (§ 4.1), training 30 models in each of the two conditions, where the number of words for each simulation is $n \sim Normal(20, 4)$.

Figure 2 shows the accuracy on all test words (All) and accuracy on test words where the final vowel is neutral (Neutral). D2L's accuracy, in both cases, is higher for the OPAQUE condition than the TRANSPARENT condition, consistent with the overall picture that humans are better at learning harmony with an opaque vowel (§ 3). D2L shows this asymmetry because, in most TRANSPARENT samples, the number of exceptions introduced by BN stems does not rise above the TP threshold (9), so D2L does not create a new representation.

No other model shows this pattern. H&L and G&G learn both kinds of harmony equally well. Thus, while Finley (2015) conjectured that the added complexity of Hayes and Londe (2006)'s distal harmony constraint might translate into difficulty learning transparent harmony, when tested on even this quite small amount of data, there is enough input to assign a weight to the distal constraint large enough for the transparent vowel to be learned. Perhaps surprisingly, even G&G, which learns to project tiers and learns its constraints, also fails to show any difference between conditions. In the OPAQUE condition, G&G consistently finds a trigram constraint that marks vowels differing in backness across another segment. This is sufficient to learn the harmony pattern. In the TRANSPARENT condition, G&G learns a similar constraint, but only specific to the harmonizing (non-neutral) vowels. G&G then projects a tier that includes only the vowels in that constraint—the non-neutral vowels. Then, on this projection, G&G learns a new constraint that marks disharmony between vowels on the tier—which excludes the transparent vowel. Thus, G&G learns transparency in conditions where humans do not.

TSLIA does not learn either harmony pattern. This indicates that there is no characteristic sample present in the data. This is true even though I collapsed irrelevant differences among segments (e.g. all consonants were mapped to the symbol C, as described in § 4.1.1), which simplifies the learning problem and in some cases leads learners of this sort to succeed at learning (Aksënova, 2020; Johnson and De Santo, 2023). Running the model without collapsing segments yields the same results.

4.3 Eventual Learning of Transparent

In the second experiment, I evaluated whether D2L and the comparison models get better at learning a transparent vowel as the amount of training exposure to words that unambiguously show the transparency of the vowel increases. This follows the same setup as the TRANSPARENT condition above,

[5] github.com/cbelth/algophon/tree/main/algophon/models

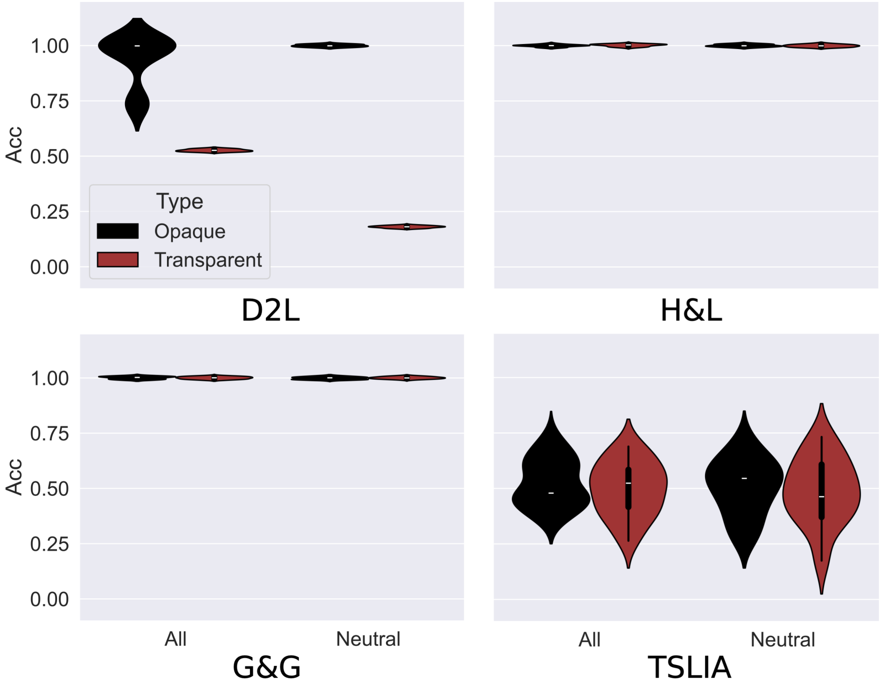


Figure 2: The distribution of accuracies (over All test words and over Neutral test words) of each model in Opaque and Transparent conditions. Only D2L shows a difference in accuracy between conditions, as humans do.

but varies two parameters: the number of BN (unambiguously transparent) types (4, 6 or 8), and the relative token frequency of those types (1x, 2x, or 5x the token frequency of non-BN types). Since the number of words for each simulation is $n \sim Normal(20, 4)$ and the choice of those n words is based on a sample weighted by token frequency, varying the relative token frequency of the BN words increases the probability that they enter into a particular learner's vocabulary. Thus, the token frequency also influences the type frequency of BN words, but in a different way. Increasing the type frequency was accomplished by replacing FN words with BN words (so the total number of words available was always 24). Combining these variations means there are 9 conditions per model. I ran 30 simulations (different seeds) for each model in each condition.

Figure 3 gives the results, where the top row of heatmaps is accuracy over all words and the bottom row is accuracy over words with neutral vowels. If a model mirrors the basic pattern of humans, who get better with transparency as exposure to BN increases, then accuracy should increase (darker colors) as the type frequency increases (rightward movement) and/or relative token frequency increases (downward movement)—in other words if more rightward and lower cells are darker. This is the case for D2L, but no other model. Increasing the prevalence of BN exceptions eventually leads D2L to form a new representation that excludes [ɛ]. H&L and G&G are dark in all cells, mirroring the above results where they learn transparent vowels when humans do not. TSLIA is again at chance across the board. D2L's performance is tied to increases in type frequency, which is consistent with arguments and evidence that type frequency, rather than token frequency, plays the primary role in the formation of linguistic generalizations (Aronoff, 1976; MacWhinney, 1978; Baayen, 1993; Elman, 1998; Pierrehumbert, 2001; Albright and Hayes, 2003; Endress and Hauser, 2011; Yang, 2016).

5 Conclusion and Discussion

Do opaque and transparent vowels do different things to a vowel harmony system? From one perspective, transparent vowels introduce non-locality that opaque vowels do not (Goldsmith, 1985; Bakovic and Wilson, 2000; Hayes and Londe, 2006; Finley, 2009). From another perspective, neither opaque nor transparent vowels change the kind of information needed to capture the harmony generalization: in both cases there is a set of seg-

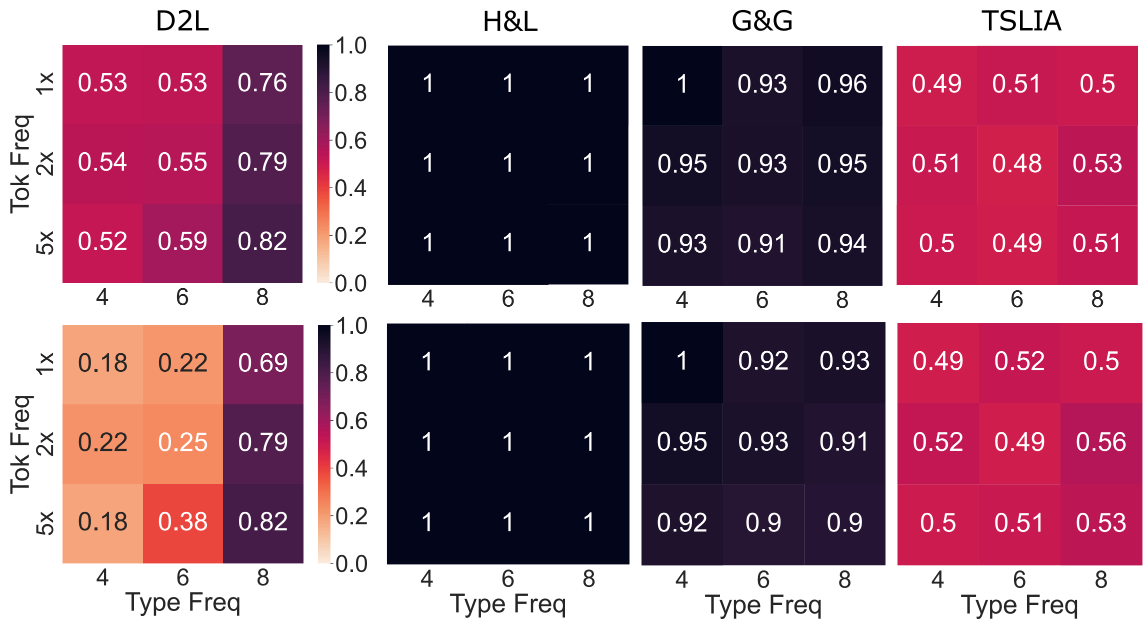


Figure 3: Heatmaps showing the accuracy of each model when trained on a vowel harmony pattern with a transparent vowel. The top row shows accuracy across all test words; the bottom shows accuracy across words where the final vowel is transparent. Matching the trend from human learners in laboratory settings would yield an accuracy gradient that increases as the type and/or token frequency of words exhibiting unambiguous transparent vowel harmony increases. D2L matches this general trend. The same cannot be said of any other evaluated models.

ments that can be projected (a tier) that renders all the dependencies local (Heinz et al., 2011; Burness and McMullin, 2019). One way to approach this question is to take the perspective of the learner. In Belth (2024), I proposed that learners construct new representations only when the ones they are currently generalizing over let them down. The results in that article demonstrated that in natural language harmony systems, this approach leads to accurate generalization to test words. Trained on a few hundred words from Turkish, where low vowels are opaque to rounding harmony, or Finnish, where [i, e] are transparent to backness harmony, D2L constructed representations that allowed for forming a successful harmony generalization. In this paper, I have demonstrated that in Finley (2015)'s setting, the same model constructs a vowel tier and only when a transparent vowel introduces enough exceptions does the model again construct a new representation, then generalizing to transparent vowels. Thus, in this proposal, there is a difference between opaque and transparent vowels—but only for a time.

Further research into the factors influencing human leaning of vowel harmony in the presence of opaque and transparent vowels—in particular children's learning and acquisition—would be of great value. For instance, D2L predicts that, if the conditions are right, there could be a stage of acquisition where learners incorrectly harmonize alternating vowels with preceding transparent vowels. In the limited number of developmental studies on the acquisition of vowel harmony systems with transparent vowels (MacWhinney, 1978; Gósy, 1989; Leiwo et al., 2006), I am not aware of reports of such errors (see Goad and Ozburn 2024 for a recent survey). However, if such a stage exists, D2L predicts it to be transient, since accumulating exceptions would lead to recursive creation of a new representation. Moreover, it is only a subset of words (BN stems in the languages discussed here) that have the potential of showing such overgeneralization. And over-application of generalizations to a particular word is influenced by the strength of the word's lexical representation, which in turn is influenced by its token frequency (Hooper, 1976; Bybee, 1985; Marcus et al., 1992; Bybee, 1995). Errors are thus more likely on low-token-frequency words, which are less represented in child speech. Consequently, identifying whether this is indeed a developmental stage would likely require studies aimed precisely at this question.

References

Alëna Aksënova. 2020. *Tool-assisted induction of subregular languages and mappings*. Ph.D. thesis, State University of New York at Stony Brook.

Adam Albright and Bruce Hayes. 2003. Rules vs. analogy in English past tenses: A computational/experimental study. *Cognition*, 90(2):119–161.

Mark Aronoff. 1976. Word formation in generative grammar. *Linguistic Inquiry Monograph*, 1.

Richard N Aslin, Jenny R Saffran, and Elissa L Newport. 1998. Computation of conditional probability statistics by 8-month-old infants. *Psychological science*, 9(4):321–324.

Harald Baayen. 1993. On frequency, transparency and productivity. In *Yearbook of morphology 1992*, pages 181–208. Springer.

Eric Bakovic and Colin Wilson. 2000. Transparency, strict locality, and targeted constraints. In *West Coast Conference on Formal Linguistics.*, pages 43–56.

Caleb Belth. 2023a. Towards a learning-based account of underlying forms: A case study in Turkish. In *Proceedings of the Society for Computation in Linguistics 2023*, pages 332–342, Amherst, MA. Association for Computational Linguistics.

Caleb Belth. 2023b. *Towards an Algorithmic Account of Phonological Rules and Representations*. Ph.D. thesis, University of Michigan.

Caleb Belth. 2024. A learning-based account of phonological tiers. *Linguistic Inquiry*, pages 1–37.

Stefan Benus and Adamantios I Gafos. 2007. Articulatory characteristics of Hungarian 'transparent'vowels. *Journal of Phonetics*, 35(3):271–300.

Phillip Burness and Kevin McMullin. 2019. Efficient learning of Output Tier-based Strictly 2-Local functions. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 78–90, Toronto, Canada. Association for Computational Linguistics.

Phillip Burness, Kevin McMullin, and Jane Chandlee. 2021. Long-distance phonological processes as tier-based strictly local functions. *Glossa: a journal of general linguistics*, 6(1).

Joan Bybee. 1985. *Morphology: A study of the relation between meaning and form*. John Benjamins, Philadelphia.

Joan Bybee. 1995. Regular morphology and the lexicon. *Language and cognitive processes*, 10(5):425–455.

Tsung-Ying Chen. 2024. The "starting-small" effect in phonology: Evidence from biased learning of opaque and transparent vowel harmony. *Language and Speech*, page 00238309241230625.

George N Clements. 1976. *The autosegmental treatment of vowel harmony*. Indiana University Linguistics Club.

George N Clements. 1980. *Vowel harmony in nonlinear generative phonology*. Indiana University Linguistics Club Bloomington.

Jeffrey Elman. 1998. Generalization, simple recurrent networks, and the emergence of structure. In *Proceedings of the twentieth annual conference of the Cognitive Science Society*, page 6. Mahwah, NJ: Lawrence Erlbaum Associates.

Ansgar D Endress and Marc D Hauser. 2011. The influence of type and token frequency on the acquisition of affixation patterns: Implications for language processing. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 37(1):77.

Sara Finley. 2009. *Formal and cognitive restrictions on vowel harmony*. The Johns Hopkins University.

Sara Finley. 2015. Learning nonadjacent dependencies in phonology: Transparent vowels in vowel harmony. *Language*, 91(1):48.

József Fiser and Richard N Aslin. 2002. Statistical learning of higher-order temporal structure from visual shape sequences. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 28(3):458.

Heather Goad and Avery Ozburn. 2024. Vowel harmony in language acquisition. In *The Oxford Handbook of Vowel Harmony*. Oxford University Press.

John Goldsmith. 1976. *Autosegmental phonology*. Ph.D. thesis, Massachusetts Institute of Technology.

John Goldsmith. 1985. Vowel harmony in Khalkha Mongolian, Yaka, Finnish and Hungarian. *Phonology Yearbook*, 2(1):253–275.

Sharon Goldwater and Mark Johnson. 2003. Learning OT constraint rankings using a maximum entropy model. In *Proceedings of the workshop on variation within Optimality Theory*, pages 111–120.

Rebecca Gómez and Jessica Maye. 2005. The developmental trajectory of nonadjacent dependency learning. *Infancy*, 7(2):183–206.

Rebecca L Gómez. 2002. Variability and detection of invariant structure. *Psychological Science*, 13(5):431–436.

Nayeli Gonzalez-Gomez, Silvana Schmandt, Judit Fazekas, Thierry Nazzi, and Judit Gervain. 2019. Infants' sensitivity to nonadjacent vowel dependencies: The case of vowel harmony in hungarian. *Journal of Experimental Child Psychology*, 178:170–183.

Mária Gósy. 1989. Vowel harmony: interrelations of speech production, speech perception, and the phonological rules. *Acta Linguistica Hungarica*, 39(1/4):93–118.

Maria Gouskova and Gillian Gallagher. 2020. Inducing nonlocal constraints from baseline phonotactics. *Natural Language & Linguistic Theory*, 38(1):77–116.

Bruce Hayes and Zsuzsa Cziráky Londe. 2006. Stochastic phonological knowledge: The case of Hungarian vowel harmony. *Phonology*, 23(1):59–104.

Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic inquiry*, 39(3):379–440.

Jeffrey Heinz, Chetan Rawal, and Herbert G Tanner. 2011. Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human language technologies*, pages 58–64.

Joan B. Hooper. 1976. Word frequency in lexical diffusion and the source of morphophonological change. In William M. Christie, editor, *Current progress in historical linguistics*, pages 96–105. North Holland, Amsterdam.

Adam Jardine and Jeffrey Heinz. 2016. Learning tier-based strictly 2-local languages. *Transactions of the Association for Computational Linguistics*, 4:87–98.

Adam Jardine and Kevin McMullin. 2017. Efficient learning of tier-based strictly k-local languages. In *Language and Automata Theory and Applications*, pages 64–76. Springer.

Jacob K Johnson and Aniello De Santo. 2023. Evaluating a phonotactic learner for MITSL-(2, 2) languages. *Society for Computation in Linguistics*, 6(1):379–382.

Bariş Kabak. 2011. Turkish vowel harmony. *The Blackwell companion to phonology*, pages 1–24.

Matti Leiwo, Pirjo Kulju, and Katsura Aoyama. 2006. The acquisition of Finnish vowel harmony. *Finnish Journal of Linguistics*, (19):149–161.

Brian MacWhinney. 1978. The acquisition of morphophonology. *Monographs of the society for research in child development*, pages 1–123.

Gary F Marcus, Steven Pinker, Michael Ullman, Michelle Hollander, T John Rosen, Fei Xu, and Harald Clahsen. 1992. Overregularization in language acquisition. *Monographs of the society for research in child development*, pages i–178.

Connor Mayer and Travis Major. 2018. A challenge for tier-based strict locality from uyghur backness harmony. In *Formal Grammar 2018: 23rd International Conference*, pages 62–83. Springer-Verlag.

Andrew Nevins. 2010. *Locality in vowel harmony*. Linguistic Inquiry Monographs. Mit Press.

Avery Ozburn, G Hansson, and Kevin McMullin. 2016. Learning vowel harmony with transparency in an artificial language. In *Talk Presented at the 2016 NOWCAM Meeting: Eugene Oregon*.

Janet Pierrehumbert. 2001. Stochastic phonology. *Glot international*, 5(6):195–207.

Catherine O Ringen and Orvokki Heinämäki. 1999. Variation in Finnish vowel harmony: An OT account. *Natural Language & Linguistic Theory*, 17(2):303–337.

Jenny R Saffran, Richard N Aslin, and Elissa L Newport. 1996. Statistical learning by 8-month-old infants. *Science*, 274(5294):1926–1928.

Jenny R Saffran, Elizabeth K Johnson, Richard N Aslin, and Elissa L Newport. 1999. Statistical learning of tone sequences by human infants and adults. *Cognition*, 70(1):27–52.

Jenny R Saffran, Elissa L Newport, Richard N Aslin, Rachel A Tunick, and Sandra Barrueco. 1997. Incidental language learning: Listening (and learning) out of the corner of your ear. *Psychological science*, 8(2):101–105.

Lynn M Santelmann and Peter W Jusczyk. 1998. Sensitivity to discontinuous dependencies in language learners: Evidence for limitations in processing space. *Cognition*, 69(2):105–134.

Kathryn D Schuler, Charles Yang, and Elissa L Newport. 2016. Testing the tolerance principle: Children form productive rules when it is more computationally efficient to do so. In *CogSci*, volume 38, pages 2321–2326.

Kathryn Dolores Schuler. 2017. *The acquisition of productive rules in child and adult language learners*. Ph.D. thesis, Georgetown University.

Rushen Shi and Emeryse Emond. 2023. The threshold of rule productivity in infants. *Frontiers in Psychology*, 14:1251124.

Charles Yang. 2016. *The price of linguistic productivity: How children learn to break the rules of language*. MIT press.

Language Learning as Codebreaking: The Key Roles of Redundancy and Locality

Richard Futrell
University of California, Irvine
rfutrell@uci.edu

Abstract

Understanding the inherent properties that render a language learnable remains a fundamental question in cognitive science and linguistics. I propose to analyze language learning as a codebreaking task, wherein the learner recovers the underlying grammar (the cryptographic key) from observed linguistic input (intercepted ciphertext). I develop a standard information-theoretic analysis of this codebreaking problem, but with a twist: in cryptography, one wants to make a code unbreakable, but in language, one wants the language to be learnable. The analysis yields three main findings: (1) Semantic redundancy—predictability of meanings given context—is necessary for language learning; (2) When learners have limited memory for sequential information, this redundancy must be local within linguistic strings; and (3) certain simple kinds of compositional languages naturally embody this kind of local semantic redundancy, enhancing their learnability. The framework shows how distributional statistics enable the learning of form–meaning mappings even when learners only observe forms.

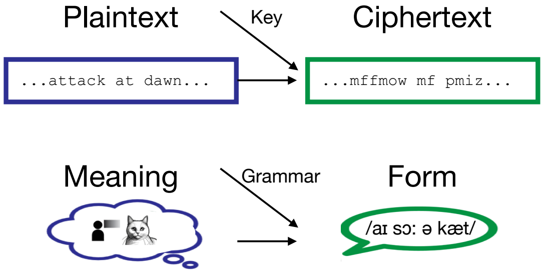


Figure 1: Parallel between language and cryptography. In cryptography (top row), a plaintext (a string) is encrypted using a secret key to form a ciphertext (another string). An attacker may determine the secret key by observing many ciphertexts; the system is designed to make this codebreaking task difficult. In language (bottom row), a meaning (in an arbitrary representational format) is expressed as a form (a string) using an unknown grammar. A learner may determine the grammar by observing forms; if the language is to be learnable, it should be structured so that this codebreaking task is easy.

1 Introduction

Theoretical models of language learning often focus on the knowledge that a human brings to the task, in the form of formal restrictions on possible grammars (Chomsky, 1965), simplicity biases (Hsu and Chater, 2010; Hsu et al., 2013), or Bayesian priors (Griffiths and Kalish, 2007; Pearl, 2023). Here I instead ask what properties of language make it learnable regardless of prior knowledge, based on a cryptanalytic approach: I consider the language learner to be a codebreaker attempting to infer a cryptographic key (the grammar of a language, which I take to include the lexicon) based on intercepted encrypted ciphertexts (linguistic input). I adapt the classic information-theoretic treatment of this codebreaking problem (Shannon, 1949) with a twist: whereas in cryptography one is interested in designing codes where the key is hard to break, here I treat language as a code that wants to be broken. The parallel language learning and codebreaking is illustrated in Figure 1.

I present three main results:

- Language learning crucially depends on *semantic redundancy* of the input.
- Given that learners have limited memory for sequences, this redundancy must be *local* within strings.
- Certain simple kinds of *compositional* languages exhibit exactly this kind of local redundancy and are more learnable as a result.

Furthermore, the cryptanalytic approach clarifies when and how semantics can be learned from distributional statistics (Harris, 1954; Mikolov et al., 2013; Merrill et al., 2021).

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 54-63.
Eugene, Oregon, July 18-20, 2025

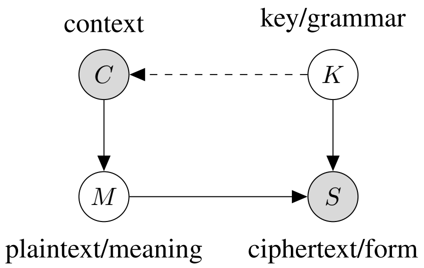


Figure 2: Probabilistic graphical model representation of the learning problem. Forms are a function of a key/grammar K and a meaning M. The learner observes context C and form S and tries to infer the key/grammar K. The learner never observes underlying meanings M. For extralinguistic context, there is no dependency of C on K. For intralinguistic context, there is such a dependency.

2 Language learning as codebreaking

Idealizing, let a **language** L_k be an injective mapping from **plaintexts/meanings** $\mathcal{M}$ to **ciphertexts/forms** which are strings drawn from a finite alphabet, parameterized by a **key/grammar** k, with each key corresponding to a unique possible mapping.[1] Let M be a random variable over meanings, K be a random variable over keys, and $S = L_k(M)$ be a random variable over forms derived by applying some language to meanings M. The context C may be extralinguistic (for example, the sensory context of a caretaker pointing to a ball before saying “ball”) or intralinguistic (for example, the words “that red” appearing before “ball”). The structure of the problem is schematized in a probabilistic graphical model in Figure 2.

The main quantity of interest for the codebreaking problem is the **leakage rate**, the amount of information that each ciphertext sample S provides about the key K. In cryptography one wants to minimize the leakage rate, but when thinking about language learnability we will be thinking about how to maximize it. Leakage rate is formally the mutual information between ciphertexts and keys given context:[2]

$$L = I[S : K \mid C]. \tag{1}$$

Each intercepted ciphertext S leaks some information about the key. The number of bits of leaked information needed to break the code is (on average) the entropy over keys $H[K]$. Leakage rate tells us how quickly the code can be broken, that is, how much ciphertext the learner must intercept before they can learn the language / determine the key, a quantity called **unicity distance** (Shannon, 1949, p. 693).

Given this analysis, there are two ways to make a language learnable.[3] The first is to set up learners to have a restricted distribution over possible grammars, thus lowering $H[K]$, the amount of leaked bits that must be gathered to break the key. The second is to increase the leakage rate, that is, to speak a language where the average form is highly informative about the key, regardless of what the prior distribution on keys looks like. I will focus on this latter aspect of language learnability.

3 Semantic redundancy

The first result is that languages are learnable to the extent that meanings are more predictable than forms. I formalize this using the notion of **semantic redundancy**, the predictability of meanings given context. I operationalize semantic redundancy using the conditional entropy of meaning given context $H[M \mid C]$, which represents the uncertainty about meaning given context: lower conditional entropy means more semantic redundancy. We will see that a language is more learnable when this quantity is small, corresponding to high semantic redundancy. Semantic redundancy may be contrasted with **formal redundancy**, the extent to which a form is predictable given context, that is the extent to which the entropy on forms $H[S \mid C]$ is not maximal.

3.1 Derivation: The importance of semantic redundancy

The first result is that there is leakage when there is more uncertainty about form than about meaning:

Proposition 1. *For extralinguistic context C, the leakage rate L is equal to formal minus semantic entropy:*

$$L = H[S \mid C] - H[M \mid C]. \tag{2}$$

[1] In cryptography the plaintext is usually also a string, but this is not necessary for the information-theoretic analysis of codebreaking. In fact, the theory does not depend on any assumptions about the nature of the set of meanings $\mathcal{M}$.

[2] I assume familiarity with the information theory concepts of entropy and mutual information. See Cover and Thomas (2006, Ch. 2) for an introduction and reference.

[3] A reviewer suggests that *iconicity* also makes a language more learnable, for example if every word is represented by an onomatopoeic form. I believe this kind of iconicity is best thought of as a (soft) restriction on the prior over keys, such that languages containing certain iconic mappings have high prior probability.

Proof. Starting with the definition of leakage and applying standard information-theoretic identities (Cover and Thomas, 2006, Ch. 2), we get

$$L = I[S : K \mid C] \tag{3}$$
$$= H[S \mid C] - H[S \mid C, K] \tag{4}$$
$$= H[S \mid C] - I[S : M \mid C, K] - H[S \mid C, K, M]. \tag{5}$$

The last term is zero because $S = L_k(M)$ is a deterministic function given knowledge of the key k, and also we have $I[S : M \mid C, K] = H[M \mid C, K]$ because languages are injective. Finally, since keys K are independent of meanings M, we have $H[M \mid C, K] = H[M \mid C]$ and we arrive at (2). □

Remark 1. The argument depends on the fact that although the learner never has access to the true underlying meanings, they do have access to a *distribution* on meanings that they think are likely to be expressed.

Remark 2. This argument corresponds to the classic result that leakage rate is a function of redundancy per character of plaintext (Shannon, 1949, p. 689), but generalized. In the current setting, the analog to plaintexts is meanings M, but these are not necessarily expressible as strings. Shannon's result still holds, except instead of being phrased in terms of characters of plaintext, the analogous quantity is characters of ciphertext given the key (appearing in Eq. 4).

Remark 3. For intralinguistic context C, we can derive a similar form for leakage,

$$L = H[S \mid C] - H[M \mid C, K], \tag{6}$$

which differs only in that the semantic entropy is conditional on the key. This is because one can only 'unlock' the semantic redundancy in the intralinguistic context to the extent that one already knows the language. The interpretation of this quantity is largely the same as for extralinguistic context.

3.2 Why does redundancy enable learning?

There are two intuitions that elucidate why it is possible to learn a form–meaning mapping when there is a low entropy on meanings given contexts.

Intuition 1: Revealed meaning. Imagine a scenario where you know exactly the single meaning $m \in \mathcal{M}$ that will be conveyed, and receive a form $s \in \Sigma^*$. Then you can filter your distribution over languages to include the mapping $m \to s$, in addition to any other updates. This scenario is the extreme case where semantic entropy $H[M \mid C] = 0$. As $H[M \mid C]$ gets smaller, learning is more and more like this scenario: low entropy over meanings means that each utterance provides partial information about the full mapping. On the other hand, if the entropy over meanings is high, then no update or only a small update is possible.

Intuition 2: Dancing men. In *The Adventure of the Dancing Men* (Doyle, 1903), Sherlock Holmes encounters messages represented as strings of dancing men of different shapes. He deduces that this is a substitution cipher, where each English letter corresponds to a certain dancing man, and breaks the code by matching the dancing men to letters based on their statistical frequency of occurrence, the letter E being the most frequent letter. In general, a substitution cipher for English plaintexts can be broken by plotting a histogram of ciphertext letter frequencies against a histogram of English letter frequencies, and finding the mapping that makes the histograms match, an approach known as **frequency analysis**. This is possible because English letters are redundant, that is, the frequency distribution over English letters is relatively low entropy.

Similarly, given some string observations and some low-entropy distribution on meanings $H[M \mid C]$, corresponding to a highly skewed histogram, one can recover the key by matching the frequencies of strings in context with the probability distribution on meanings in those contexts. On the other hand, if the entropy of meanings $H[M \mid C]$ is high, then both the form frequencies and the meaning distribution will be close to flat, and so the histogram-matching approach will either not yield a unique solution, or will only work after intercepting a very large number of forms.

Distributional learning In distributional learning, one learns language entirely on the basis of frequency of occurrence and co-occurrence with context in the input. Distributional learning is a successful approach to modeling aspects of child language acquisition (Saffran et al., 1996) as well as developing computational representations of word meanings (Mikolov et al., 2013; Pennington et al., 2014). The result above clarifies why distributional learning works even when a learner never observes meanings directly (compare Bender and Koller, 2020): because intra- and extralinguistic contexts are informative about meaning, and thus can stand in as a proxy for meaning in an information-theoretic sense.

If language lacked semantic redundancy of this kind—that is, if $H[M \mid C]$ were maximal—then distributional learning would be impossible, as we would have $H[S \mid C] = H[M \mid C]$ and leakage $L = 0$. In fact, this corresponds to the notion of **perfect secrecy** in the cryptography setting (Shannon, 1949, §10), and optimal codes such as Huffman codes (Huffman, 1952), which minimize redundancy by design, also have minimal leakage. On the other hand, as long as the entropy of meanings $H[M \mid C]$ is not maximal (either due to context, or simply because the distribution on meanings is non-uniform), then we have nonzero leakage $L > 0$ and the learner will be able to get some information about the key.

3.3 Cognitive and linguistic significance

There are two linguistically significant interpretations of this result, depending on whether one thinks of the context C as extralinguistic or intralinguistic.

If C is extralinguistic, then the result shows the importance of the speaker's choice of which meanings to express in which contexts. Examples would include a child's caretaker pointing to a ball before saying "ball"—thus creating a context C which is highly predictive about the intended meaning M—or the caretaker choosing to name objects already present in the immediate environment, thus pedagogically choosing *meanings* M to fit the context C. Cognitively, the result requires that the child is able to infer communicative intent from context, at least to some extent, and more generally has some sense of what meanings are more or less likely. Learning is possible when meaning is low-entropy for the learner.

If C is intralinguistic, then the result shows the importance of the language itself being semantically redundant, as a function of both its grammatical structure and usage choices of the speaker. An utterance such as "My favorite vegetable is ..." provides semantic redundancy by predicting certain semantic features of the following word (provided one has already worked out the meaning of "vegatable"). Languages with grammatical cues to semantic features, such as Bantu languages with rich noun class systems, provide similar information through grammatical means. Intralinguistic semantic redundancy corresponds to the familiar experience of being able to guess the meaning of an unknown word in context, for example when reading.

3.4 The role of formal redundancy

An interesting wrinkle is that *formal* redundancy is not helpful for learning in this highly idealized setting: leakage is upper bounded by the formal entropy $H[S \mid C]$. This means that, when the *form* S of some linguistic input is highly predictable from context, this *reduces* the amount of information that the input provides to a learner.

The role of formal redundancy and its relationship with semantic redundancy must be interpreted carefully. Formal redundancy does not simply mean that a form is predictable, it means that a form is predictable *on average across the learner's key distribution*. Effectively, when the learner has narrowed down the keys to some subset, and a form is totally predictable under all those keys, then there is formal redundancy without semantic redundancy, because observing the form is totally unsurprising.

Formal redundancy without semantic redundancy can arise from, for example, phonotactic constraints. For example, suppose that a language has phonotactics where every front vowel is followed by only front vowels, that is, it has vowel harmony; and suppose that a learner is aware of the concept of vowel harmony and has narrowed their space of possible languages/keys only to those that respect vowel harmony. Then when a front vowel occurs in the context of a front vowel, it is formally redundant: it is uninformative about *anything*, including the meaning.

4 Locality: Learning with noise

The argument above establishes that a learnable language must have semantic redundancy, but tells us nothing about the structure of that redundancy. Next I consider learners whose memory or attention for sequences is noisy, such that their observations effectively consist of contiguous substrings rather than full strings. Such noisy memory is characteristic of human children (Cowan et al., 1999; Gathercole et al., 2004; Luna et al., 2004). In this setting, I find that languages are more learnable when their intralinguistic redundancy is *local*, that is, when the meaning of a character or word is predictable given nearby characters or words.

4.1 Derivation: Effect of noise on learning

I now assume that with probability e, the context C is unavailable to the learner, with $L(e)$ being the leakage rate as a function of the context erasure rate e. The idea is that a learner with limited memory or attention might find themselves processing part of a string without knowledge of its context.

In order to understand how the leakage changes as a function of noise rate e, one can calculate the derivative of $L(e)$ with respect to e:

Proposition 2. *For extralinguistic context C, the derivative of leakage with respect to context erasure rate* e *is equal to the formal minus semantic mutual information:*

$$\frac{\partial}{\partial e} L(e) = I[S : C] - I[M : C]. \tag{7}$$

Proof. Let $\tilde{C}$ represent the random variable over noisy context, equal either to a true context or to a special erasure symbol E not in the support of C. The leakage as a function of erasure rate $L(e)$ comes out to

$$\begin{aligned} L(e) &= H[S \mid \tilde{C}] - H[S \mid \tilde{C}, K] && (8)\\ &= H[S \mid C] - H[S \mid C, K] && (9)\\ &\quad + eI[S : C] - eI[S : C \mid K] \\ &= H[S \mid C] - H[M \mid C] && (10)\\ &\quad + eI[S : C] - eI[M : C]. \end{aligned}$$

The derivative of (10) with respect to e is (7). □

Remark 4. The analogous result for intralinguistic context is

$$\frac{\partial}{\partial e} L = I[S : C] - I[M : C \mid K], \tag{11}$$

paralleling the intralinguistic version of Prop. 1.

The result means that as a context becomes more likely to be unavailable to the learner, the learnability of the language goes up in proportion to the formal redundancy contributed by that context, and down in proportion to the semantic redundancy contributed by that context. Intuitively, if the learner has no access to context, then the semantic redundancy contributed by context cannot help. In terms of language learnability, the upshot is that languages should be configured so that helpful semantically redundant context is likely to be available in practice: that is, somewhere in the string where it is not likely to be erased.

4.2 Locality from noise

Consider now a scenario where a learner takes in a string incrementally and, at each position, has some probability of randomly forgetting (or otherwise ignoring) the string prefix up to that point. This represents a learner who either has noisy memory for the sequence context, or who has had a lapse of attention and is starting to process a string somewhere in the middle. Then the learner effectively has perceptual *intake* (in the sense of Pearl, 2023) consisting of contiguous substrings, rather than full strings.

In that case, if there is some helpful semantic redundancy between two nonlocal parts of a string, then this redundancy is unlikely to help the learner, since the learner is unlikely to get a large enough substring to encompass all parts. On the other hand, semantic redundancy between local parts of the string is more likely to be available. The upshot is that for a language to be learnable under these circumstances, it must have information locality (Futrell and Hahn, 2022): any helpful semantic redundancy should be expressed in *local* parts of a form, so that a learner with noisy memory or attention who is only receiving contiguous substrings as input is able to detect that redundancy and learn from it.

The idea of local semantic redundancy is related to the concept of **diffusion** from cryptanalysis (Shannon, 1949, pp. 708–709). Diffusion is a desirable property for cryptographic ciphers, where the redundancy in the plaintext is dissipated into long-range correlations involving many parts of the ciphertext, so that a codebreaker must intercept and analyze a very large quantity of contiguous ciphertext in order to detect the redundancy and exploit it. For learnability, human languages should do the opposite of diffusion: they should be set up so that semantic redundancy is detectable without considering large amounts of context.

5 Simulations

The considerations above suggest that for languages to be learnable, (1) languages must have semantic redundancy, and (2) if there is noisy memory for sequence context, languages should configure strings so that semantically redundant parts are local. Here I demonstrate this result by simulating learning of some very simple languages which differ in their levels of redundancy, in the locality of that redundancy, and in the level of noise under

Meaning →	ⒽⒽⒽ	ⒽⒽⓉ	ⒽⓉⒽ	ⒽⓉⓉ	ⓉⒽⒽ	ⓉⒽⓉ	ⓉⓉⒽ	ⓉⓉⓉ
Compositional 1	aaa	aab	aba	abb	baa	bab	bba	bbb
Compositional 2	bbb	abb	bab	aab	bba	aba	baa	aaa
Holistic 1	aab	bbb	bba	aba	baa	bab	aaa	abb
Holistic 2	abb	bbb	bab	baa	aab	aba	aaa	bba

Table 1: Example languages for the coinflip world, used in simulations. Possible meanings (coinflip outcomes) are on the columns. In the 'compositional' languages, each character corresponds to an individual coin, as indicated by color. In the holistic languages, there is no such correspondence.

which learning takes place. In line with the formal results, I find that semantic redundancy facilitates learning, and that in the presence of noise this redundancy must be local. Furthermore, I show how local redundancy obtains when languages are compositional in the sense that individual characters or local groups of characters (that is, words or morphemes) correspond to independent components of meaning.

5.1 Setup

I simulate ideal learners who start with an initial uniform distribution over keys/languages, observe (noisy) sample forms one at a time, and update their distribution on keys using Bayes' rule (Bayes, 1763).

Source As the probability distribution over meanings, I consider a very simple world consisting of two or three weighted coinflips, for a total of $2^2 = 4$ or $2^3 = 8$ possible outcomes/meanings. The first coin has weight b for heads, where I vary the weight b in order to vary the entropy of meanings $H[M]$—more biased coins yield lower-entropy distributions which should facilitate learning. The second and third coins have weights $b+0.1$ and $b+0.2$ respectively. If the coins did not have different weights, then the language would be unidentifiable for the learner, because the learner would never be able to identify which characters in a form correspond to which coins.

Languages I first consider languages where forms consist of binary strings of length 3, which are either compositional or not, in the sense that individual characters in the forms may or may not correspond to the underlying coinflips. These languages are categorized with examples in Table 1. I also consider redundant languages where forms consist of binary strings of length 4 and meanings consist of two coinflips. These languages are based on the Compositional 1 language in Table 1, and are either locally redundant (for example, a meaning ⒽⓉ is encoded as aabb) or nonlocally redundant (for example, the same meaning is encoded as abab). In all conditions, the learner's set of possible languages/keys is the set of all possible injective mappings from meanings to binary strings of the appropriate length.

Learning and noise In each step of learning, a learner observes a single (noisy) sample of a form, and updates their probability distribution on meanings exactly following Bayes' rule. Noisy observations are generated by sampling a form, splitting it into contiguous substrings, and uniformly choosing one of those substrings. The splitting is done by flipping a coin with probability e at each character of the string; if the outcome is heads, the string is split at that point. I vary the parameter e in experiments. The condition $e = 0$ corresponds to no noise. The condition $e = 1$ yields to a learner who only ever sees a single character of input based on a sampled string, corresponding to maximally noisy memory for intralinguistic context.

Evaluation I evaluate learning in terms of **key entropy**, the posterior entropy over keys given data observed so far at each timestep. Lower key entropy indicates the learner has less uncertainty about the language. The main feature of interest is the rate at which this entropy decreases.

I would like to emphasize that for all conditions in these simulations, the key entropy will eventually approach zero with enough observations: that is, learning is ultimately possible for all the languages considered here. They will differ, however, in their rates of learning.

5.2 Analysis of languages

The compositional languages in Table 1 have semantic redundancy local to each individual character. This is because the meaning of each character corresponds to one coinflip, and thus the semantic entropy for a single character is bounded: it cannot exceed the entropy of its corresponding single

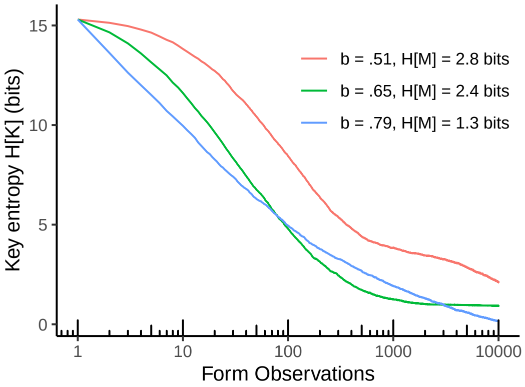


Figure 3: Learning curves (average over 10,000 runs) for different levels of semantic entropy, with no noise. Curves show key entropy $H[K]$ as a function of the number of forms observed (similar to Shannon, 1949, Fig. 6). Key entropy decreases more rapidly when semantic entropy is low. Curves are the same for all languages in Table 1.

coinflip. This redundancy is local in the sense that it does not depend on context and cannot be destroyed by erasure noise. On the other hand, in the holistic languages, each character corresponds to a *mixture* of different coins, which will generally have a higher entropy (thus less semantic redundancy) than the distribution of a single coin. Furthermore, there will be nonlocal correlations among the characters within the string, representing nonlocal semantic redundancy which is in danger of being missed due to noise. This observation is in line with the idea that noncompositional languages very generally create undesirable long-term correlations within forms (Futrell and Hahn, 2024).

The locally redundant variant of the compositional language extends this idea so that redundancy is local to a *pair* of adjacent characters. The helpful semantic redundancy in this adjacent pair is unlikely to be disrupted by noise, and thus learning curves are favorable. On the other hand, in the nonlocally redundant language, the redundancy is nonlocal, highly likely to be disrupted by noise, and so the learning curves are less favorable.

5.3 Results

Learning curves without noise ($e = 0$) by semantic entropy are shown in Figure 3, which demonstrates that learning is indeed faster when semantic entropy is lower. The language used for this simulation is Compositional 1 from Table 1, but this does not matter: in this setting, all injective languages will produce equivalent curves when there is no noise.

Learning curves under varying levels of noise are shown in Figure 4. Here we find that the compositional languages yield faster learning, as expected, because their semantic redundancy is local and not likely to be disrupted by noise. The difference between compositional and holistic languages gets bigger as the noise rate increases. Learning curves for the explicitly redundant languages are shown in Figure 5. Languages with local redundancy are faster to learn, while languages with nonlocal redundancy are slower.

6 Discussion and Related Work

I emphasize that I have considered learners who never directly observe meaning, and who have no *prior* bias towards any language over another; nor is any language 'simpler' than any other for the learners. The fact that certain languages are learned more rapidly is rather a function of their semantic redundancy and information locality, which enables learning in the presence of noisy memory or attention for sequences, in a way that is independent of the learner's prior distribution over languages.

Distributional learning This work provides a theoretical understanding of when it is possible to learn a form–meaning mapping from observations of form alone, and thus justifies distributional approaches to semantics and language learning (Harris, 1954; Erk, 2010), both in the context of language technologies (Mikolov et al., 2013), and as a strategy for child learners (Saffran et al., 1996; Erickson and Thiessen, 2015). The results are consistent with Merrill et al.'s (2024) finding that corpus statistics encode entailment relations under the assumption that speakers are redundant, and I believe the notion of local semantic redundancy is likely related to Merrill et al.'s (2021) notion of semantic transparency, which is a precondition for distributional learning of semantics.

Language acquisition The model shows how language can be acquired when context provides partial information about meanings, and thus it provides a generalized idealized version of the cross-situational learning model of lexicon acquisition (Siskind, 1996; Hendrickson and Perfors, 2019), in which a child encounters a word across multiple contexts until they can identify the word with a single meaning by a process of elimination. The re-

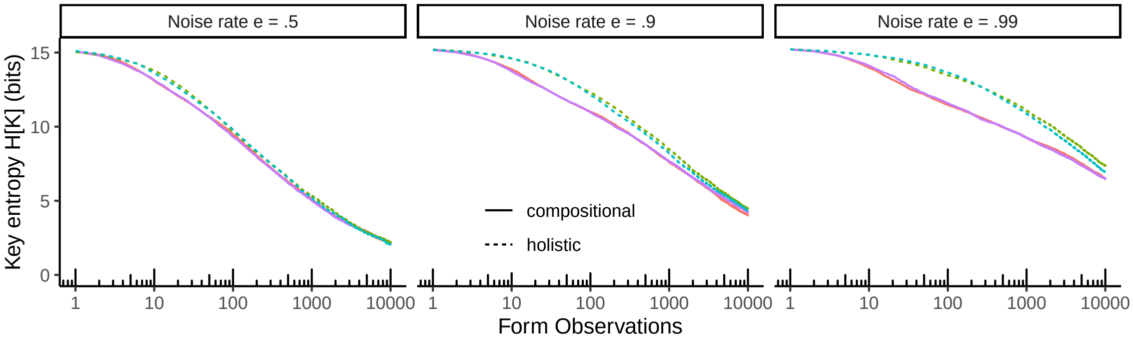


Figure 4: Learning curves for different levels of noise e, for a source with a fixed $b = .75$ (average over 10,000 runs). Curves show key entropy $H[K]$ as a function of the number of forms observed. Key entropy decreases more rapidly for the compositional languages, where semantic redundancy is local. It increases more slowly for the holistic languages where semantic redundancy is spread out among characters of the form. The difference between compositional and holistic languages is heightened for increased noise rates.

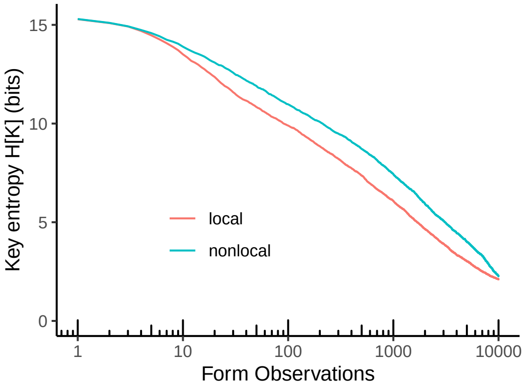


Figure 5: Learning curves for locally redundant and nonlocally redundant languages (see text) under noise at rate $e = .9$, for coinflip heads probability $b = .79$ (average over 1000 runs). Key entropy decreases more rapidly for the locally redundant languages.

sults about the importance of low semantic entropy are in line with the finding that children learn word meanings better given low-entropy input (Lavi-Rotbain and Arnon, 2019). The results on noise and locality show how cognitive constraints, such as maturational constraints on working memory, can imbue learners with a bias toward the kinds of structures found in language (Newport, 1990; Mita et al., 2025).

Unsupervised machine translation This work bears a notable similarity to models of how one can learn to translate between languages without seeing parallel texts (Cao et al., 2016), or how one might decode unknown communication systems such as those used by whales, where the nature of the meanings being expressed is unknown and possibly unknowable (Goldwasser et al., 2023). The current approach to language learning can be seen as inducing an unsupervised translation system from meanings (represented in some unknown mental form) to forms (represented as observable strings).

Language evolution Approaches to modeling language evolution by iterated learning have yielded the result that languages will generally reflect learners' prior distribution on languages (Griffiths and Kalish, 2007; Kirby et al., 2014). In contrast, I find a learning bias (toward locally redundant languages) as a function of the noisy nature of learners' intake, independent of the prior. This bias can be seen as arising from the learners' likelihood function rather than the prior, and it manifests in the *rate* of learning, not in its initial or asymptotic states. Under noise, locally redundant languages can be learned to a higher degree of confidence from fewer samples.

While humans may have innate prior knowledge of what grammars/keys are possible, the question remains of why that prior knowledge is what it is. For example, if humans' prior knowledge can be characterized by a constraint that languages must be compositional in a certain way, the question is why that constraint rather than another. The considerations above provide a potential explanation, by showing how learning biases can emerge independently of learners' priors. One could imagine a population of learners with flat priors, who end up with local compositional languages due to general memory limitations, as discussed in Section 4. Then over generations of evolutionary time, the population can evolve to incorporate these biases as innate prior knowledge.

7 Conclusion

I have presented a model of language learning based on ideas from cryptanalysis, in which a learner observes only forms and infers the underlying language, the mapping from hidden meanings to forms. Whereas in cryptanalysis one is concerned with making codes unbreakable, here I considered what properties of languages make them *breakable*. I found that languages with local semantic redundancy—the opposite of cryptographic diffusion, and corresponding to a kind of compositionality—are more learnable in this setting, even for learners without prior biases toward such languages. The model shows how learning is possible as long as the learner has some prior knowledge of their interlocutor's likely communicative intent.

The analytical and modeling approach taken here provides a useful new angle on language learning which can be applied to test hypotheses about how learning works, how properties of language affect learnability, and how the learner's hypothesis space on languages could be structured to enable rapid learning. More broadly, I believe that this cryptography-inspired analysis of language learning offers a fresh perspective and set of analytical tools that can be used to approach the language learning problem. Cryptanalysis is a well-developed and rich field of science and engineering. The analysis here shows that it may contain useful ideas for linguistics and language acquisition.

Acknowledgments

I thank Greg Scontras, Lisa Pearl, Connor Mayer, the UCI QuantLang Collective, and the three anonymous reviewers for helpful comments.

Code availability

Code to reproduce the simulations and figures is available at `http://github.com/langprocgroup/locallearning`.

References

Thomas Bayes. 1763. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S., communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S. *Philosophical Transactions of the Royal Society of London*, 53:370–418.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198.

Hailong Cao, Tiejun Zhao, Shu Zhang, and Yao Meng. 2016. A distribution-based model to learn bilingual word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1818–1827, Osaka, Japan.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA.

Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory*. John Wiley & Sons, Hoboken, NJ.

Nelson Cowan, Lara Nugent, Emily M. Elliott, Igor Ponomarev, and John Scott Saults. 1999. The role of attention in the development of short-term memory: Age differences in the verbal span of apprehension. *Child Development*, 70(5):1082–1097.

Arthur Conan Doyle. 1903. The adventure of the dancing men. *The Strand Magazine*, 26(156):603–617.

Lucy C. Erickson and Erik D. Thiessen. 2015. Statistical learning of language: Theory, validity, and predictions of a statistical learning account of language acquisition. *Developmental Review*, 37:66–108.

Katrin Erk. 2010. What is word meaning, really? (And how can distributional models help us describe it?). In *Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics*, pages 17–26, Uppsala, Sweden. Association for Computational Linguistics.

Richard Futrell and Michael Hahn. 2022. Information theory as a bridge between language function and language form. *Frontiers in Communication*, 7:657725.

Richard Futrell and Michael Hahn. 2024. Linguistic structure from a bottleneck on sequential information processing. *arXiv preprint arXiv:2405.12109*.

Susan E. Gathercole, Susan J. Pickering, Benjamin Ambridge, and Hannah Wearing. 2004. The structure of working memory from 4 to 15 years of age. *Developmental Psychology*, 40(2):177–190.

Shafi Goldwasser, David Gruber, Adam Tauman Kalai, and Orr Paradise. 2023. A theory of unsupervised translation motivated by understanding animal communication. In *Advances in Neural Information Processing Systems*, volume 36, pages 37286–37320. Curran Associates, Inc.

Thomas L. Griffiths and Michael L. Kalish. 2007. Language evolution by iterated learning with Bayesian agents. *Cognitive Science*, 31(3):441–480.

Zellig S. Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Andrew T. Hendrickson and Andrew Perfors. 2019. Cross-situational learning in a Zipfian environment. *Cognition*, 189:11–22.

Anne S. Hsu and Nick Chater. 2010. The logical problem of language acquisition: A probabilistic perspective. *Cognitive Science*, 34(6):972–1016.

Anne S. Hsu, Nick Chater, and Paul Vitányi. 2013. Language learning from positive evidence, reconsidered: A simplicity-based approach. *Topics in Cognitive Science*, 5(1):35–55.

David A. Huffman. 1952. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101.

Simon Kirby, Thomas L. Griffiths, and Kenny Smith. 2014. Iterated learning and the evolution of language. *Current Opinion in Neurobiology*, 28C:108–114.

Ori Lavi-Rotbain and Inbal Arnon. 2019. Children learn words better in low entropy. In *Proceedings of the 41st Annual Meeting of the Cognitive Science Society*, pages 631–637.

Beatriz Luna, Krista E. Garver, Trinity A. Urban, Nicole A. Lazar, and John A. Sweeney. 2004. Maturation of cognitive processes from late childhood to adulthood. *Child Development*, 75(5):1357–1372.

William Merrill, Yoav Goldberg, Roy Schwartz, and Noah A. Smith. 2021. Provable limitations of acquiring meaning from ungrounded form: What will future language models understand? *Transactions of the Association for Computational Linguistics*, 9:1047–1060.

William Merrill, Zhaofeng Wu, Norihito Naka, Yoon Kim, and Tal Linzen. 2024. Can you learn semantics through next-word prediction? The case of entailment. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2752–2773, Bangkok, Thailand. Association for Computational Linguistics.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Masato Mita, Ryo Yoshida, and Yohei Oseki. 2025. Developmentally-plausible working memory shapes a critical period for language acquisition. *arXiv preprint arXiv:2502.04795*.

Elissa L. Newport. 1990. Maturational constraints on language learning. *Cognitive Science*, 14(1):11–28.

Lisa Pearl. 2023. Computational cognitive modeling for syntactic acquisition: Approaches that integrate information from multiple places. *Journal of Child Language*, 50(6):1353–1373.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Jenny R. Saffran, Richard N. Aslin, and Elissa L. Newport. 1996. Statistical learning by 8-month-old infants. *Science*, 274(5294):1926–1928.

Claude E. Shannon. 1949. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715.

Jeffrey M. Siskind. 1996. A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, 61(1-2):39–91.

A LSTM language model learns Hindi-Urdu case-agreement interactions, and has a linear encoding of case

Satoru Ozaki and Rajesh Bhatt and Brian Dillon
University of Massachusetts Amherst
{sozaki,bhatt,bwdillon}@umass.edu

Abstract

Much evaluation work in the literature shows that neural language models seem capable of capturing syntactic dependencies in natural languages, but they usually look at relatively simple syntactic phenomena. We show that a two-layer LSTM language model trained on 250M morphemes of Hindi data can capture the relatively complex interaction between case and agreement in Hindi-Urdu, at an accuracy of 81.17%. Furthermore, we show that this model encodes case-marking linearly, implementing a geometrically intuitive and interpretable syntactic processing mechanism. We also show that this model doesn't calculate agreement extremely eagerly, as case information seems to be persistent over time as a sentence unfolds. This is surprising given LSTMs autoregressive and recurrent nature, which should exert an incremental processing pressure onto our model.

1 Introduction

Neural language models trained for engineering purposes tend to show human-like behavior when evaluated on certain benchmarks constructed to test their understanding of syntactic properties of certain natural languages. These results are quite significant, because they show that neural networks capture syntactic dependencies that target latent hierarchical structures even when they are trained on an objective as simple as next-word prediction, which doesn't provide any explicit signal about hierarchical structure. However, these benchmarks often only target relatively simple grammatical phenomena, such as English subject-verb number agreement. Thus, we don't know if language models really learn the full range of complex phenomena featured in various natural languages. Another problem concerns interpretability: when these language models display human-like behavior, what kind of computation underlies their such performances? Understanding the expressibility and the computation implemented by language models is empirically important for assessing whether they are viable models of grammar and sentence processing. In this paper, we show a LSTM language model (Gulordava et al., 2018) trained on Hindi data predicts the correct agreement form of a participial verb correctly 81.17% of the time, and encodes ergative and accusative case in a subspace of its hidden layer vectors in a way that makes representations for sentences containing each of these case-markers linearly separable from those that don't contain each case-marker. Our results suggest that a LSTM language model is not only capable of learning the relatively complex interaction between case and agreement in Hindu-Urdu, but also encodes case-marking information in a geometrically intuitive and interpretable fashion. We think this work points to a direction for future work in which we can compare language models with different architectures in how they represent and compute with case.

This paper is organized as follows. In Section 2, we describe relevant work. We discuss two groups of methods: those for evaluating language models' ability to learn syntactic properties of natural languages, and those for understanding the representations and computations tacitly implemented by language models.

In Hindu-Urdu, verb agreement targets different arguments depending on their case-marking patterns, making it a relatively complex agreement pattern and a good testing ground for evaluating language models' ability to capture syntactic dependencies. We describe the Hindi-Urdu facts in more detail in Section 3, and the training and evaluation procedures as well as evaluation results in Section 5. Despite the modest model size and training setup, the language model performs reasonably well, predicting the correct gender agreement 81.17% of the time.

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 64-73.
Eugene, Oregon, July 18-20, 2025

In the rest of the paper, we investigate the nature of the computation that underlies our language model's decent performance. This investigation is carried out from two perspectives, which we describe in Section 6. The first one concerns how the language model represents case. We set forth a very specific hypothesis, which is that the model provides a linear encoding of case. If this were true, the model implements a highly interpretable syntactic processing mechanism. The second perspective concerns the memory usage. The computation underlying our language model could be *eager* and *Markovian*, making use of the subject's case information as soon as it is processed, after which this piece of information no longer has any bearing on the predicted gender agreement. Alternatively, it could be *lazy* and *memory-intensive*, storing the subject's case information in its intermediate representations, using it just-in-time as the model predicts a gender agreement marker. In the latter case, subject case information is used long after it has processed the subject.

We carry out the investigation using linear classifier probes and causal intervention techniques. These methods, as well as our results, are described in Section 7. We find positive evidence that the language model provides such a linear encoding for the presence/absence of ergative and accusative case. Our results also align with a lazy characterization of the language model's underlying computation. We conclude in Section 8.

2 Background and related work

There has been much interest in evaluating language models' understanding of grammatical phenomena, a practice sometimes known as *targeted syntactic evaluation* (Marvin and Linzen, 2018). LSTM language models have been evaluated on various syntactic phenomena, including subject-verb agreement (Linzen et al., 2016; Bernardy and Lappin, 2017; Kuncoro et al., 2018; Gulordava et al., 2018), negative polarity item licensing (Jumelet and Hupkes, 2018; Marvin and Linzen, 2018) and filler-gap dependencies (Chowdhury and Zamparelli, 2018; Chaves, 2020; Da Costa and Chaves, 2020; Wilcox et al., 2024). They show various levels of success on each phenomenon.

Much research also seeks to *interpret* language models, i.e., understand their internal mechanisms that grant them their performances. One popular approach in this area is to *probe* language models for representations of certain kinds of grammatical information. Typically, this involves extracting the intermediate representations from a language model produced for certain linguistic expressions, and using them to train and evaluate a shallow classifier that predicts some relevant grammatical information associated with these expressions. For example, Tenney et al. (2019) show that BERT representations can be used to predict syntactic categories of and dependency relations between constituents in English.

A common criticism of probing is that it involves training; thus a positive result can't necessarily be attributed to the language model. There are ways to overcome this problem. For example, probing with weak linear classifiers allows one to conclude that the relevant grammatical information is encoded by the language model as a subspace, allowing a geometrically intuitive interpretation of the language model's inner workings. Further, by *counterfactually intervening* the language model's representations using the classifier probe's weights and checking if the intervention affects the language model's inference process, one can check if the language model is actually using the grammatical information the way it is encoded as suggested by the classifier probe. A recent line of work incorporates both of these aspects; for example, Hao and Linzen (2023) find a linear encoding of number in a subspace of BERT's contextualized representations for English, and show that causal intervention in this subspace affects BERT's performance on subject-verb number agreement tasks.

Agreement is a classic example of a syntactic dependency that targets hierarchical structure; a lot of interpretability work has focused on LSTM language models' learning of agreement. Linzen et al.'s (2016) pioneering work shows that LSTMs are capable of predicting English number agreement as a classification task, on which they are trained with explicit supervision. Gulordava et al. (2018) show that LSTM language models naturally learn to predict number agreement correctly in Italian, English, Hebrew and Russian. Lakretz et al. (2019) argue that two units in Gulordava et al.'s (2018) language model track number, which means LSTM language models implement genuine syntactic processing mechanisms.

3 Case and agreement in Hindi-Urdu

In Hindi-Urdu, the participial main verb and any auxiliary agree with the structurally most prominent argument of the verb that is not case-marked overtly (Bhatt, 2005). The subject is more structurally prominent than the object. The overt case marker for subjects is *-ne*, which we will call *ergative case*. The overt case marker for objects is *-ko*, which we will call *accusative case*. For example, when the subject is not marked ergative, the verb and auxiliary agree with the subject no matter whether the object is marked accusative or not (1). This agreement is coded on an aspectual morpheme that immediately follows the verb stem.

(1) Rahul kitaab(-ko) paṛh-taa
Rahul[**M**] book[F](-ACC) read-HAB;MSG
thaa
be[PST;**MSG**]
'Rahul used to read a/the book.'

When the subject is marked ergative, agreement targets the object if the object is not marked accusative (2).

(2) Rahul-ne kitaab paṛh-ii thii
Rahul[M]-ERG book[**F**] read-PFV;F be[PST;**FSG**]
'Rahul had read a book.'

When both arguments are overtly case-marked, agreement targets neither argument. The result is default masculine agreement, shown in (3), where there are no masculine arguments.

(3) Sita-ne kitaab-ko paṛh-aa
Sita[F]-ERG book[F]-ACC read-PFV;MSG
thaa
be[PST;**MSG**]
'Sita had read the book.'

While case controls agreement in Hindi-Urdu, case itself is controlled by independent factors. The subject receives ergative case iff its verb is transitive and in the perfective aspect. The object receives accusative case iff it is specific or definite.

4 Current study

As described in the previous section, Hindi-Urdu features a more complex verbal agreement system than subject-verb agreement systems found in languages like English, making it an interesting challenge for language models to learn. In the rest of this paper, we train a LSTM language model on Hindi data, and address the following two research questions concerning this model. First, how well does the model learn the case-agreement interaction in Hindi-Urdu (Section 5)? Second, if learning is successful, how does the model compute agreement using case information (Sections 6–7)? In particular, we employ causal intervention techniques to answer the second question.

5 Training and evaluation

5.1 Training

The training data for our language model comes from the Hindi Wikipedia (Foundation) and the Hindi data from the CC-100 corpus (Conneau et al., 2020; Wenzek et al., 2020), both taken from the Hugging Face website. The data mostly consists of unromanized Devanagari. We perform unsupervised morphological segmentation with Morfessor 2.0 (Smit et al., 2014), which reduced our vocabulary size from 2.4M to 146K. We then discarded all sentences longer than 80 morphemes and converted all morphemes except the most frequent 30000 to a designated UNK(nown) token, giving us about 246M non-UNK tokens. We follow a train:dev:test split of 7:1:2.

We train Gulordava et al.'s (2018) LSTM language model. Due to the limited size of our training data, we decided to train a LSTM language model rather than a Transformer. Gulordava et al. show that their LSTM language models predict Italian number agreement across long-distance dependencies at near-human performance. The architecture of the model is a two-layer LSTM with an embedding size and hidden layer size of 650. We follow the set of hyperparameters that gave Gulordava et al. their best validation set perplexity, which we detail in Appendix A. Our test set perplexity is 47.17, comparable to Gulordava et al.'s results.

5.2 Evaluation

We artificially generate an evaluation dataset intended to test our language model's ability to predict gender agreement correctly. Each data point is a pair $\langle s, \gamma \rangle$ where s is a sentence prefix and γ is a gender label. The sentence prefix s consists of a subject, an object and a verb stem, and should be continued with an aspectual morpheme that shows gender agreement. The correct gender is encoded by the label γ. The data points are manipulated by three conditions: whether or not the subject is marked ergative, whether or not the object is marked accusative, and the genders of the subject and object, which are always different. Table 1 illustrates the kinds of data points generated for each combination of conditions. We combinatorially generate 320K data points. Most sentence prefixes in the data set are semantically nonsensical, an intended effect; we want the model to rely only on structural properties of the data, not semantic ones.

Evaluation proceeds as follows. For each data point with sentence prefix s and correct gender γ, we compare the conditional probability of the masculine and feminine singular forms of the following four aspectual morphemes given the context s:

(4) a. HAB: habitual (M: ता, F: ती)
b. INF: infinitival (M: ना, F: नी)
c. PFVC: perfective morpheme that begins with the consonant य (M: या, F: यी)
d. PFVV: perfective morpheme that doesn't begin with a consonant (M: ◌ा, F: ◌ी)

Within each aspect, the form corresponding to gender γ should be higher than the form for the incorrect gender. Accuracy is aggregated over the dataset for each aspect. Incorporating results from multiple aspectual forms gives us a more comprehensive evaluation with more generalizable results, unlike previous evaluation work on English subject-verb number agreement that only focuses on one auxiliary pair, e.g. *is*/*are*.

However, it can be misleading to compare accuracy across items or conditions within each aspect, because certain aspectual morphemes are incompatible with certain items and conditions. For example, whether a verb takes the PFVC or the PFVV morpheme in the perfective is lexically specified; a verb takes PFVC iff its stem ends in a vowel (e.g. सजा *sajā*, but not भेज *bhej*). Ergative marking results only in the perfective. A language model with adequate knowledge of Hindi-Urdu may reasonably assign equally low probabilities to the masculine and feminine PFVC forms of the verb भेज *bhej*, and to the masculine and feminine PFVC forms of the verb सजा *sajā* when the subject is not ergative, because all of these forms are ungrammatical. This would result in a low accuracy for PFVC forms.

To address this, we also calculate a form of accuracy that incorporates all aspects. Specifically, for each data point, we compare the probability summed over the masculine forms of all four aspects with the probability summed over the feminine forms of all four aspects. Intuitively, the summation represents marginalization over aspect, allowing us to compare the probability of the two genders directly. We call the accuracy aggregated over the dataset this way *general accuracy*. Table 2 reports the by-aspect and general accuracy for our language model, broken down by subject and object case-marking as well as the correct gender to show agreement for, i.e., the gender label γ.

Additionally, in Table 2, we report the sensitivity index d' for all three case patterns that doesn't result in default masculine agreement. We calculate d' as $z(\text{hits}) - z(\text{FA})$, where z is R's qnorm function, hits is the proportion of true masculine examples correctly predicted masculine, and FA (false alarm) is the proportion of true feminine examples incorrectly predicted masculine. Thus, d' quantifies the language model's sensitivity to the agreement contrast after factoring out any general biases towards masculine or feminine morphemes the model may have.

Among the four aspects, the habitual aspect (HAB) gives the best results, with a high accuracy of 82.69 and a sensitivity index d' of 1.84. In comparison, the other aspects have a slightly above-chance performance. Recall that general accuracy and d' are calculated by comparing the marginal probabilities of the masculine vs. feminine forms, where marginalization is summation over aspects. General accuracy is 81.18 and d' is 1.73, a decent performance. For comparison, Gulordava et al. (2018) train models with the same architecture on Italian, English, Hebrew and Russian data, and evaluate their models using two subject-verb agreement tasks. They report accuracies in the range 67.5–95.2. The results suggest that our language model has reasonably understood the case-agreement interaction in Hindi-Urdu.

6 Characterizing the language model's underlying computation

We see that our language model has learned the case-agreement interaction in Hindi-Urdu to some extent. What kind of computation could our language model be performing in order to determine agreement?

To frame this question more specifically, let's consider what forms this computation can take. A correct Hindi-Urdu agreement computation can be thought of generally as a process that takes case information as input and returns the agreement target as output. For example, it can be modelled as the simulation of a finite-state machine illustrated in Figure 1, where case determines the transitions and the accepting states determine which argument the agreement should target. The simulation keeps

Genders	Cases	Data point $\langle s, \gamma\rangle$	Glossed example for s
MASC. subject, FEM. object	∅,∅	$\langle \mathit{NP}_{\text{M}}\ \mathit{NP}_{\text{F}}^{\neg A}\ V, \text{M}\rangle$	कुमार एक माता छोड़ Kumar[M] one mother[F] leave
	∅,ACC	$\langle \mathit{NP}_{\text{M}}\ \mathit{NP}_{\text{F}}^{A}\ \mathit{acc}\ V, \text{M}\rangle$	कुमार एक माता को छोड़ Kumar[M] one mother[F] ACC leave
	ERG,∅	$\langle \mathit{NP}_{\text{M}}\ \mathit{erg}\ \mathit{NP}_{\text{F}}^{\neg A}\ V, \text{F}\rangle$	कुमार ने एक माता छोड़ Kumar[M] ERG one mother[F] leave
	ERG,ACC	$\langle \mathit{NP}_{\text{M}}\ \mathit{erg}\ \mathit{NP}_{\text{F}}^{A}\ \mathit{acc}\ V, \text{M}\rangle$	कुमार ने एक माता को छोड़ Kumar[M] ERG one mother[F] ACC leave
FEM. subject, MASC. object	∅,∅	$\langle \mathit{NP}_{\text{F}}\ \mathit{NP}_{\text{M}}^{\neg A}\ V, \text{F}\rangle$	सीता एक पिता छोड़ Sita[F] one father[M] leave
	∅,ACC	$\langle \mathit{NP}_{\text{F}}\ \mathit{NP}_{\text{M}}^{A}\ \mathit{acc}\ V, \text{F}\rangle$	सीता एक पिता को छोड़ Sita[F] one father[M] ACC leave
	ERG,∅	$\langle \mathit{NP}_{\text{F}}\ \mathit{erg}\ \mathit{NP}_{\text{M}}^{\neg A}\ V, \text{M}\rangle$	सीता ने एक पिता छोड़ Sita[F] ERG one father[M] leave
	ERG,ACC	$\langle \mathit{NP}_{\text{F}}\ \mathit{erg}\ \mathit{NP}_{\text{M}}^{A}\ \mathit{acc}\ V, \text{M}\rangle$	सीता ने एक पिता को छोड़ Sita[F] ERG one father[M] ACC leave

Table 1: Data point templates for each combination of conditions, with examples. In the **Cases** column, ∅ means no overt case-marking; e.g., ∅,ACC means non-overtly marked subject, ACCusative-marked object. In the **Data point** column, each sentence prefix s is described as the right-hand side of a rewrite rule. Uppercase variables are non-terminals: NP_γ stands for a singular noun phrase with gender γ, NP_γ^A specifically stands for one that may be ACC-marked, i.e., specific or definite, $\mathit{NP}_\gamma^{\neg A}$ specifically stands for one that may not be ACC-marked, i.e., not specific or definite. V stands for a verb stem.

ERG?	ACC?	Correct	HAB		INF		PFVC		PFVV		General	
			Acc	d'	**Acc**	d'	**Acc**	d'	**Acc**	d'	**Acc**	d'
−	−	M	74.18	1.18	57.86	0.33	32.11	0.32	93.76	0.28	78.00	1.04
−	−	F	70.35		55.38		78.36		10.52		60.56	
−	+	M	98.96	3.17	88.84	1.12	72.08	0.97	99.97	2.03	99.53	3.16
−	+	F	80.57		46.09		64.97		8.24		71.22	
+	−	M	95.25	2.55	59.52	1.14	56.10	1.41	97.05	1.62	84.42	1.96
+	−	F	81.02		81.44		89.63		39.49		82.82	
+	+	M	83.30		68.38		79.68		99.88		91.60	
Average			82.69	1.84	65.27	0.77	68.00	1.08	66.80	1.17	81.18	1.73

Table 2: Accuracy and d' for our language model evaluated on the case-agreement dataset.

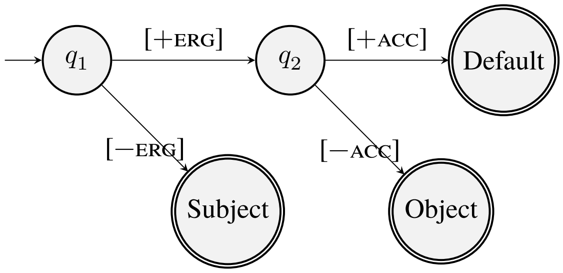


Figure 1: Agreement computation as a finite-state machine.

track of the current state, and follows transitions depending on subject and object case.

In order for the language model to implement such a simulation, it needs to represent case information somehow. But exactly how does it represent case? We take a very specific hypothesis to this question: our language model linearly encodes case in a subspace of its hidden layer vectors. That is, there is a subspace for ergative case, such that the hidden layer vectors for sentences with an ergative-marked subject are linearly separable from those for sentences with an non-ergative-marked subject when both sets of vectors are projected onto this subspace. In other words, we can use an ensemble of linear binary classifiers to predict the presence of ergative marking in a sentence from its hidden layer vector representation. The same applies to accusative case. Under this hypothesis, our language model implements a highly interpretable syntactic processing mechanism.

Aside from representations of the input, we can also consider other aspects of this computation. One dimension along which we can characterize alternative forms of computation is memory usage. This places an *eager* and *Markovian* computation on one end of a spectrum, and a *lazy* and *memory-intensive* computation on the other end. These two computations differ in how soon they advance the simulation as they process linguistic input. As soon as an eager and Markovian computation processes case information, it advances the simulation by following the corresponding transition. A lazy and memory-intensive computation would store the subject and case information, and performs the entire simulation in one fell swoop when it reaches the verb stem, just in time before it needs to compute agreement. Where is our language model's underlying computation located along this eager/lazy spectrum?

In the next section, we use linear classifier probes and causal intervention techniques to investigate whether our language model encodes case linearly, and how eager/lazy it is at advancing the simulation.

7 Investigating the language model's underlying computation

For our first investigation, we first explore the hypothesis that the language model linearly encodes the presence/absence of each case-marking as a subspace in its hidden layers. To do this, we first use a method known as *iterative nullspace projection* (INLP) to find three sets of orthonomal basis vectors that identify a potential case subspace; two for ergative, and one for accusative. We then rerun the evaluation described in Section 5.2, but intervening on the subject and object representations, reflecting them onto the "opposite side" of the case subspaces, effectively making the representation of a case-marked argument not case-marked, and that of a non-case-marked argument case-marked. We check how effective the intervention is by measuring how intervention affects the language model's performance. An effective intervention suggests the subspace identified by INLP really is how the language model encoding case.[1]

7.1 Method: intervention

Intervention is a process that takes three things as input: a vector $x \in \mathbb{R}^d$, which is a representation produced by our language model, a set of orthonormal basis vectors $\mathbb{B} = b_1, \cdots, b_k \in \mathbb{R}^d$, which identifies a subspace that encodes case, and an *intensity parameter* $\alpha \geq 1$. First, for each $j = 1, \cdots, k$, calculate λ_j, the scalar projection of x onto b_j with $\lambda_j = x^\top b_j$. Then, return the intervened vector $x' \in \mathbb{R}^d$, calculated as $x' = x - \alpha \sum_{j=1}^{k} \lambda_j b_j$. The interpretation of x' depends on α. When $\alpha = 1$, x' is the projection of x onto the nullspace of the case subspace; x' then represents x but with all case information removed. When $\alpha = 2$, x' is the reflection of x onto the opposite side of the case subspace; x' then inverts the case information of x. For example, if $\mathbb{B}$ represents the ergative subspace, and x represents an ergative-marked argument, then x' represents the same argument as x except it's non-ergative-marked. Any $\alpha > 2$ pushes x' further in the opposite case direction, intensifying the effect of the intervention.

[1] Our description of intervention and INLP largely follows Hao and Linzen's (2023) presentation.

7.2 Method: iterative nullspace projection

To perform intervention with respect to a case subspace, we first need a set of orthonormal basis for that subspace. Iterative nullspace projection (INLP) (Dufter and Schütze, 2019; Ravfogel et al., 2020) is a supervised method to help us find the bases for a subspace of interest. We describe INLP for identifying the ergative subspace; the same process works for the accusative subspace. First, we designate a training split of the evaluation dataset, and run the language model on each sentence prefix $s^{(i)}$ of the training split to obtain a hidden layer vector $h^{(i)}$ at some position of interest. Each $h^{(i)}$ is paired with a binary label $c^{(i)}$ representing the presence/absence of ergative case in that sentence prefix. Then, we train a linear classifier to predict $c^{(i)}$ from $h^{(i)}$. The normalized weights of the classifier, a vector in $\mathbb{R}^d$, is taken to be the first basis b_1. For each additional jth basis we'd like to find, we train another linear classifier the same way, except we preprocess the input $h^{(i)}$ by intervening it with the first $j-1$ bases and intensity $\alpha = 1$, removing the ergative case information captured by the first $j-1$ bases. We train each classifier using gradient descent, which guarantees that the new classifier weight b_j is a weighted sum of the preprocessed inputs $h^{(i)}$. Since the preprocessing projects each $h^{(i)}$ onto the nullspaces of the first $j-1$ bases, b_j is guaranteed to be orthogonal to all of $b_1, \cdots b_{j-1}$.

7.3 Evaluation with causal intervention: is case encoded linearly?

We perform a 50-fold cross validation on the evaluation dataset, with a training split of 6.4K data points in each fold. For the ergative subspace, we run INLP on hidden layer vectors obtained from two positions: one set after processing the subject, and another after processing the object. For the accusative subspace, we run INLP on hidden layer vectors obtained after processing the object. This gives us three sets of bases: one for the post-subject ergative subspace, one for the post-object ergative subspace, and one for the post-object accusative subspace.

The remaining 313.6K data points in each fold is used for evaluation. We re-run the evaluation described in Section 5.2, while performing causal intervention with respect to each one of the three case subspaces at the appropriate location. For example, for the post-object ergative subspace, we feed each sentence prefix into our language model, and pause once the model processes the object. We intervene the hidden layer vectors with respect to the post-object ergative subspace using some intensity α, and resume model inference using the intervened hidden layer vectors, effectively flipping the presence/absence of ergative marking. We compare the agreement performance of the language model before and after the intervention to see how successful the intervention was. We use sensitivity index (d') to quantify model performance. The results are shown in Figure 2 for ergative intervention and Figure 3 for accusative intervention. We present the results for $\alpha = 5$ just as Hao and Linzen (2023) did, noting that lower values for α doesn't change our results qualitatively.

Let's first consider ergative intervention. We believe ergative case information should be the most recoverable at the post-subject position; hence in this section, we only look at the results of the post-subject ergative intervention. In the [-ERG,-ACC] condition, agreement should target the subject. A successful ergative intervention should assimilate this to the [+ERG,-ACC] condition, where agreement should target the object. Indeed, we see that the agreement performance flips to the opposite prediction, as d' drops below zero. In the [-ERG,+ACC] condition, agreement should target the subject. A successful ergative intervention assimilates this to the [+ERG,+ACC] condition, which requires default agreement. This should be reflected as chance performance, which is exactly what we see in our results. Finally, in the [+ERG,-ACC] condition, agreement should target the object. A successful intervention assimilates this to the [-ERG,+ACC] condition, where agreement should target the subject. However, our ergative intervention only drives the agreement performance to near-chance level, not exactly reversing the agreement predictions.

Let's turn to accusative intervention. In the two [-ERG] conditions, a successful accusative intervention shouldn't affect agreement computations, because agreement should always target the subject if it isn't ergative-marked. Indeed, our intervention doesn't change the agreement predictions qualitatively, as it remains above chance in both conditions. In the [+ERG,-ACC] condition, agreement targets the object. A successful accusative intervention should cause agreement to fall back to default masculine. However, our intervention keeps the agreement above chance, which means agreement is still targetting the object.

Thus, we have found positive evidence that our

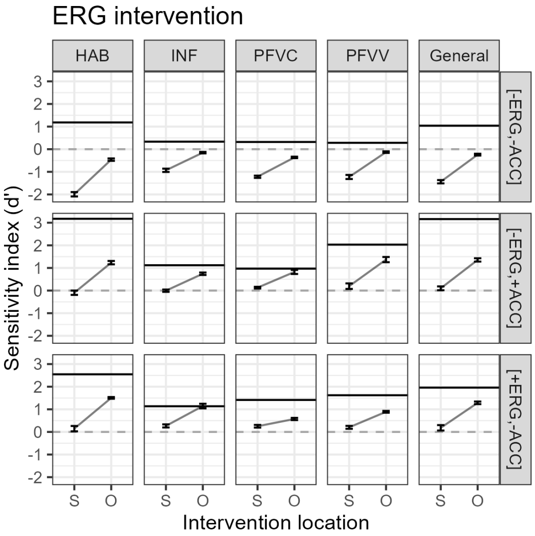


Figure 2: Ergative intervention results. The two intervention locations are post-**S**ubject and post-**O**bject. The light dashed line is drawn at $d' = 0$, indicating chance performance. The dark solid line indicates the original performance of the language model before intervention. Error bars indicate one standard error average across cross validation.

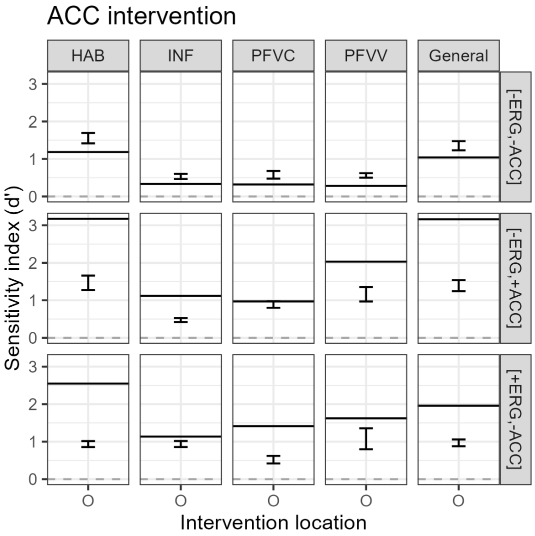


Figure 3: Accusative intervention results. The intervention location is post-**O**bject.

language model uses a linear encoding of ergative and accusative case marking, and uses this encoding to calculate agreement.

7.4 Is agreement computation eager or lazy?

For our second investigation, we check whether our language model aligns more with an eager or a lazy characterization of agreement computation. We suggest that looking at the effectiveness of the post-object ergative intervention may give us a clue, because it should only be effective in a lazy, but not an eager, computation. An eager computation would use the ergative case information to advance the simulation as soon as it processes the subject, discarding that information, while a lazy computation would store the ergative case information until it sees the verb. Looking at Figure 2 again, we observe that post-object ergative intervention is still effective, although the magnitude of the intervention effect is smaller than post-subject intervention. This suggests our language model isn't computing agreement in a purely eager way.

Although this by itself is a very weak conclusion, we think that the general method of causal interventions with respect to linear encodings we pursue here can be extended in interesting ways to help us better understand the underlying computation of language models. For example, we plan to perform the same analysis we describe in this paper to Transformers. We think the autoregressive and recurrent nature of the LSTM architecture create an incremental processing pressure that encourages performing computations on the fly, while Transformers aren't subject to this pressure. Thus, we expect Transformers to show signs of a lazier computation than our LSTM language model.

8 Conclusion

In this paper, we train a LSTM language model on Hindi data and show that it has learned case-agreement interactions in Hindi-Urdu, predicting correct gender agreement 81.17% of the time. We further show that our language model has learned to encode case information in a low-dimensional subspace of its hidden layer vectors, where case-marked arguments are linearly separable from non-case-marked arguments. In addition, our model uses case information encoded this way as part of its agreement computation. Preliminary evidence also suggests that our language model doesn't calculate agreement extremely eagerly, as our causal intervention methods reveal that case information seems to be persistent over time as the language model processes a sentence. The general method described in this paper can be adopted to study interesting phenomena concerning case and agreement in other languages.

Limitations

We see two limitations in our work, which both concern interpreting our causal intervention results. The first limitation is that we don't know how much of the effectiveness of our case interventions is meaningful. For example, Figure 2 shows that post-subject ergative intervention in the [−ERG,−ACC] condition decreases general d' by about 2.5. Can all of this 2.5 point decrease be attributed to successful ergative intervention? For example, if we had performed multiple post-subject interventions, each time with respect to a set of randomly generated orthonormal basis vectors, and observed a d' decrease in the range 1.5 to 3, then our ergative intervention result wouldn't be meaningful, since just any intervention would affect d' in a similar way. We plan to add a comparison between our current results and intervention with respect to random bases in a future version of this paper.

The second limitation is that we presently offer no way of quantifying how lazy or eager our language model's underlying computation is. This would be possible if we know how effective we would expect post-object ergative intervention to be under a fully lazy and a fully eager computation. While a fully lazy computation should result in equal effectiveness between post-object and post-subject ergative intervention, we don't know how effective a fully eager computation should be. In the future, we hope to consider alternative ways of quantifying the eagerness/laziness of our language model's underlying computation.

References

Jean-Phillipe Bernardy and Shalom Lappin. 2017. Using deep neural networks to learn syntactic agreement. *Linguistic Issues in Language Technology*, 15(2).

Rajesh Bhatt. 2005. Long distance agreement in Hindi-Urdu. *Natural Language & Linguistic Theory*, 23:757–807.

Susana Béjar and Milan Rezac. 2009. Cyclic Agree. *Linguistic Inquiry*, 40(1):35–73.

Rui Chaves. 2020. What don't RNN language models learn about filler-gap dependencies? In *Proceedings of the Society for Computation in Linguistics 2020*, pages 1–11, New York, New York. Association for Computational Linguistics.

Shammur Absar Chowdhury and Roberto Zamparelli. 2018. RNN simulations of grammaticality judgments on long-distance dependencies. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 133–144, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jillian Da Costa and Rui Chaves. 2020. Assessing the ability of transformer-based neural models to represent structurally unbounded dependencies. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 12–21, New York, New York. Association for Computational Linguistics.

Philipp Dufter and Hinrich Schütze. 2019. Analytical methods for interpretable ultradense word embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1185–1191, Hong Kong, China. Association for Computational Linguistics.

Wikimedia Foundation. Wikimedia downloads.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. 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 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Sophie Hao and Tal Linzen. 2023. Verb conjugation in transformers is determined by linear encodings of subject number. *Preprint*, arXiv:2310.15151.

Jaap Jumelet and Dieuwke Hupkes. 2018. Do language models understand anything? on the ability of LSTMs to understand negative polarity items. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 222–231, Brussels, Belgium. Association for Computational Linguistics.

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Melbourne, Australia. Association for Computational Linguistics.

Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. 2019. The emergence of number and syntax units in LSTM language models. 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 11–20, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

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.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256, Online. Association for Computational Linguistics.

Shauli Ravfogel, Yoav Goldberg, and Tal Linzen. 2019. Studying the inductive biases of RNNs with synthetic variations of natural languages. 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 3532–3542, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. *Preprint*, arXiv:1905.06316.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

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

Annie Zaenen, Joan Maling, and Höskuldur Thráinsson. 1985. Case and grammatical functions: the Icelandic passive. *Natural Language & Linguistic Theory*, 3(4):441–483.

A Hyperparameters

Our LSTM had a hidden layer size of 650. We trained with an initial learning of 20, gradient clipping with a maximum L2 norm of 0.25, truncated backpropagation through time with window size 35, a dropout rate of 0.2 for 40 epochs. Whenever the validation set perplexity doesn't improve in a new epoch, the learning rate is scaled by a factor of 0.25.

Modeling sentence polarity asymmetries: Fuzzy interpretations in a possibly wonky world

Muxuan He **Elsi Kaiser** **Khalil Iskarous**
University of Southern California
{muxuanhe, emkaiser, kiskarou}@usc.edu

Abstract

Negation is an important aspect of human language and reasoning. Prior work has proposed that positive- and negative-polarity sentences exhibit a number of asymmetries. This paper focuses on two of them: (i) Regarding cost, marked forms like negation are known to elicit more production cost than the unmarked positive polarity, and (ii) regarding pragmatic inference, the negative polarity is said to presuppose the prominence of its positive-polarity counterpart, but not the other way around. We present novel empirical evidence regarding these two asymmetries and offer one of the first formalizations of these asymmetries within the Rational Speech Act (RSA) framework. We show that existing extensions of the standard RSA model, e.g., soft semantics and common ground update, while not originally proposed to address sentence polarity asymmetries, can nonetheless be applicable to these phenomena.

1 Introduction

As one of the most influential cognitive models of pragmatics, the Rational Speech Act model (RSA; Frank and Goodman, 2012) formalizes the recursive reasoning involved in language use and communication. See formulas (1) - (4) for a formal definition of the standard RSA model:

$$P_{L0}(s|u) \propto [\![u]\!](s) \cdot P(s) \quad (1)$$

$$[\![u]\!](s) \in \{0, 1\} \quad (2)$$

$$P_{S1}(u|s) \propto exp(\alpha\ (ln\, P_{L0}(s|u) - Cost(u)) \quad (3)$$

$$P_{L1}(s|u) \propto P_{S1}(u|s)\ \cdot P(s) \quad (4)$$

This model centers on a pragmatic listener, $P_{L1}(s|u)$, who infers the intended state s from an utterance u by reasoning about a pragmatic speaker, $P_{S1}(u|s)$, who selects utterances based on their utility U. This speaker derives informativeness (how much an utterance reduces uncertainty about the intended meaning or referent) from a literal listener, $P_{L0}(s|u)$, who interprets u deterministically as true or false ($[\![u]\!](s) \in \{0, 1\}$) and factors in the cost of u, $Cost(u)$. The speaker is modeled as a SoftMax-optimal agent choosing utterances to best convey s. Both listeners apply Bayesian inference to update beliefs over states from the prior, $P(s)$, which serves as the shared common ground (Stalnaker, 1978, 2002).

The RSA model and its close extensions successfully cover a wide range of pragmatic phenomena (see Degen, 2023; Scontras et al., 2021 for a review), including those involving negation, such as indirect politeness and negative strengthening (e.g., *not bad* vs. *not amazing* in Yoon et al., 2020), projective content that survives negation (Qing et al., 2016), and presupposition triggering (Warstadt, 2022). However, the use of RSA to specifically address the pragmatic consequences of sentence polarity asymmetries has received less attention. Theoretical work on negation (e.g., Jakobson, 1963; Givón, 1978; Horn, 1989) suggests that positive and negative polarities show (at least) two asymmetries, which we refer to as Asymmetry Hypotheses 1 and 2:

- Asymmetry Hypothesis 1: Marked forms like negation are typically realized using more complex structures and longer linguistic forms, which are known to elicit higher production cost than their unmarked counterparts; and
- Asymmetry Hypothesis 2: Negation presupposes that its positive-polarity counterpart is relevant or prominent in the common ground, not the other way around.

In this paper, we aim to (i) empirically test the pragmatic consequences of the two asymmetry hypotheses and to (ii) characterize the empirical patterns associated with two types of asymmetry within the RSA framework.

The first asymmetry is closely linked to the trade-off between informativeness and cost that the pragmatic speaker in RSA must consider. Given that a pragmatic speaker aims to maximize informativeness and minimize cost, the standard RSA model predicts that a negative utterance is less likely to be produced than a similarly informative positive-polarity utterance, i.e., when the states they refer to have similar prior probabilities. Consider part-whole relations as a concrete example. Assuming that situations like *The house doesn't have a bathroom* and *The house has a* ***ball****room* have similar prior probabilities (see below for details on a norming study of state priors), utterances describing these situations should be similarly informative. However, when the standard RSA model (in particular, the pragmatic speaker) penalizes higher-cost utterances, the negative utterance yields a lower utility and is therefore less likely to be produced.

The second asymmetry regarding presupposition accommodation is closely related to common ground update. Assuming that negation presupposes the probability of its positive-polarity counterpart, a negative utterance requires that this positive counterpart be either part of the common ground or can be accommodated. If it is not already common ground knowledge, listeners must accommodate the presupposition before the negative utterance can successfully update the common ground with the negated information. Thus, if a speaker says *The house doesn't have a ballroom*, then in principle the negative utterance presupposes the possibility of *The house has a ballroom*. However, since *ballroom* is not a typical part of *house*, the listener must accommodate this atypical part-whole relation before the negative utterance can be deemed pragmatically motivated and smoothly integrated into common ground.

Utterance choices can be easily probed by asking naïve participants how likely they are to mention certain things. In contrast, directly asking whether a negative utterance presupposes the possibility of its positive-polarity counterpart is less likely to yield interpretable results. To probe this second asymmetry, we instead asked participants to rate the typicality of the whole entity under discussion (e.g., *house*, see more details in Experiment 2 in Section 3.2).

As we show in Section 3, (i) the empirical data patterns are more complex than those predicted by either hypothesis, (ii) while the standard RSA model aligns with the predictions of Hypothesis 1, it fails to account for our findings, and (iii) the standard RSA model lacks a mechanism for common ground updating such that it can't capture Hypothesis 2, let alone explain the observed data. In light of this, we extend the standard model to better capture our empirical findings.

2 Related Work

The standard RSA model (formalized in (1)-(4)) tends to idealize the key components–such as common ground and the literal listener–that are, in practice, subject to uncertainty in real-world communication. Before delving into the empirical findings and our extended RSA models, we review relevant work on common ground update and soft semantics (as opposed to deterministic semantics).

2.1 Common ground update in RSA

Degen et al. (2015) observed that the single prior mechanism in the standard RSA model predicts no scalar implicature in a *some*-utterance that introduces a high-prior event, e.g., *Some marbles sank into water*, while both theoretical observations (Geurts, 2010) and empirical data (Degen et al., 2015) suggest that the scalar implicature is, in fact, strong. To solve this issue, Degen et al. proposed a complex prior, $P(s|w)$ in (5), which determines the world (wonky vs. normal) based on the variable *wonkiness*, *w*. In their ***w*RSA model** (see (5) – (8)), the pragmatic listener, P_{L1} $(s, w|u)$, jointly infers the actual state and the world wonkiness.

$$P(s|w) \propto \begin{cases} 1 & if\ wonky\ world \\ P(s) & if\ normal\ world \end{cases} \tag{5}$$

$$P_{L0}(s|u,w) \propto [\![u]\!](s) \cdot P(s|w) \tag{6}$$

$$P_{S1}(u|s,w) \propto exp(\alpha\ (ln\, P_{L0}(s|u,w) - Cost(u)) \tag{7}$$

$$P_{L1}(s,w|u) \propto P_{S1}(u|s,w)\ \cdot P(s|w) \cdot P(w) \tag{8}$$

This model predicts that, when observing a *some*-utterance that introduces a high-prior event, the pragmatic listener backs off to the wonky world where the event has a lower prior probability. This adjustment makes the *some*-utterance a more reasonable utterance choice for the speaker. Degen et al.'s study shows that this extended model fits the empirical data much better than the basic model, in terms of updating both state and world priors.

Kravtchenko and Demberg (2022b), using the core ideas from the *w*RSA model to predict *atypicality inferences* in redundant descriptions of habitual events, found that low-utility utterances led listeners to infer that the habituality of an agent's actions was lower than typically expected.

However, as Cremers et al. (2023) point out, Degen et al. (2015)'s implementation of the *w*RSA model deviates from strict Bayesian reasoning. Instead of directly using the empirically obtained prior distribution over world states in the pragmatic listener's belief of common ground, the model assigns a weighted combination of two worlds: one uniform ('wonky world') and one empirical (representing 'normal world'), which contaminates the so-called 'observation'. Therefore, Cremers et al. (2023) replaced *P(s|w)* with *P(s|normal world)* for the literal listener. See (9) for the modification that we adapted from Cremers et al. (2023):

$$P_{L1}(s,w|u) \propto P_{S1}(u|s,w) \cdot P(s|normal) \cdot P(w) \quad (9)$$

Degen et al.'s proposal of a complex prior inspired more work on the joint inference of common ground and state (Qing et al., 2016; Warstadt, 2022) that involve another approach, namely, Question under Discussion (QUD; Roberts, 1996/2012). By inferring a pragmatic speaker's question under discussion, the pragmatic listener finds a way to rationalize utterances.

For the present study, we want to start with the approach of complex prior, for which our empirical data provide a meaningful test ground. However, this does not exclude QUD as a future direction.

2.2 Soft semantics in RSA

The literal listener's model in the vanilla RSA model and most of its variants interprets an utterance with a deterministic Boolean semantics. Using the examples from Degen et al. (2020), the utterance "small" assigns a probability of 0 to the referent 'big red ball' (false) and the referent 'big blue ball' (false) and assigns a probability of 1 to the referent 'small blue ball' (true), in a finite set consisting only of these three objects.

"Small ball" is the optimal utterance for a listener to most efficiently identify the 'small blue ball', but in natural production, speakers are often redundant, producing "small blue ball" instead. To address this and other empirical-modeling discrepancies with referential expressions, Degen et al. (2020) introduced soft semantics—a continuous semantics—into the RSA model.

Continuing with the examples from Degen et al. (2020), the soft semantics of the utterance "small" can assign a probability of .48 to the 'small blue ball' and a probability of .26 to both the 'big blue ball' and the 'big red ball', reflecting flexibility in literal meaning. Such fuzzy (i.e., vague in the sense of fuzzy logic, Zadeh, 1978) interpretations can be simply represented as follows:

$$[\![u]\!](s) \in [0\,,1] \subset \mathbb{R} \quad (10)$$

The literal interpretation is no longer restricted to a binary 'true' vs. 'false' but instead ranges from 0 to 1 in a continuous manner. Regarding the implementation of this continuous semantics, probabilities of literal meanings are decided during model fitting, e.g., using optimization techniques such as Maximum Likelihood Estimation (Degen et al., 2020), or by plugging in pre-normed data when applicable (Yoon et al., 2020). In addition to Degen et al. (2020), the model of the literal listener can also be modified by introducing lexical uncertainty to the lexicon (Bergen et al., 2012).

Degen et al. (2020)'s approach can be interpreted as introducing noise to literal meaning. Relatedly, Bergen and Goodman (2015)'s noisy-channel RSA introduces noise to the transmission of utterance itself that affects literal meaning as well: The received utterance may differ from the intended utterance at the string level. Kravtchenko and Demberg (2022b) adapted the noisy-channel RSA to model the effects of framing on atypicality inferences, showing that emphasis (e.g., via exclamation punctuation) strengthens these inferences. They argue that with emphasis redundant utterances are less prone to misremembering or being ignored, and thus more likely to trigger pragmatic inferences.

In the case of negation, soft semantics might be able to capture both types of noises, namely fuzzy interpretations of negative utterances and their potentially noisy transmission. This is suggested by various prior observations regarding negation: (i) Theoretically, negation is said to presuppose the existence of the negated (Horn, 1989), (ii) empirically, negative sentences trigger the activation of both the negated representation (e.g., *door-not open*) and the negative representation (e.g., *door-open*) (Kaup et al., 2006), and (iii) negation impacts memory in that negative situations can be misremembered as their positive counterparts (Maciuszek & Polczyk, 2017; Cornish & Wason, 1970).

3 Sentence Polarity Asymmetries

We collected utterance choice preferences in Experiment 1 to test Hypothesis 1 and the standard RSA model. We collected typicality ratings in Experiment 2 to test Hypothesis 2. As previewed

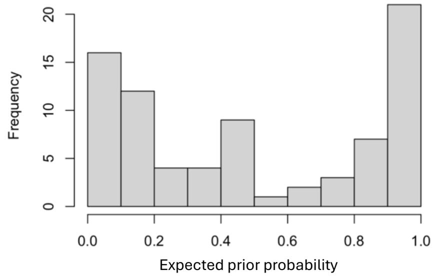


Figure 1: **Norming study.** Histogram of expected values of each smoothed prior distribution

earlier, the results of the experiments reveal more nuanced sentence polarity asymmetries than can be fully captured by either of the two asymmetry hypotheses or the standard RSA model.

3.1 Norming of the state prior

In both experiments, our stimuli were sentences describing real-world part-whole relations such as *house-garage* and their negative forms *house-no garage*. To test how prior probabilities of these part-whole relations influence utterance likelihood and sentence interpretation in the standard RSA model and human data, we first conducted a norming study. This norming study (n=57) measured prior probabilities of 81 part-whole pairs.

The pairs consisted of 27 whole entities and three part entities for each whole entity. On each trial, participants saw two words: the whole entity in capitals (e.g., *CLASSROOM*) and the part entity in lower case (e.g., *stove*). Participants gave their ratings on a slider scale (0-100%) to answer questions about *state prior probability*, e.g., how likely they think a stove is part of or seen in a classroom. The percentage rating distributions for each pair were smoothed using a nonparametric density estimation method suited for ordinal categorical variables with the np package in R (Hayfield & Racine, 2008), following Degen et al. (2015). This non-parametric smoothing method is used in all experiments reported here to handle outliers in our relatively small samples, while preserving the ordinal nature of the rating data.

As Figure 1 shows, the data have a wide range of coverage while somewhat oversampling the high and low ends of probability. This is ideal for generalizing findings across levels of state priors.

3.2 Informativeness-cost trade-off

Experiment 1 (n=52) measured utterance likelihoods of individual part-whole relations being explicitly mentioned. On each trial, participants read a two-sentence sequence followed by a question. The first sentence is a lead-in that introduces the 'whole' entity, e.g., *Emma visited a friend's house yesterday*. The second sentence states a fact about what the place has (i.e., the 'part' entity), in either positive or negative polarity (*The house has a bathroom* or *The house doesn't have a bathroom*). Each participant saw an equal amount of positive and negative-polarity items. For each item, participants rated utterance likelihood, e.g. *How likely do you think it is that Emma would mention that?* Participants gave their ratings on a slider scale (0- 100%).

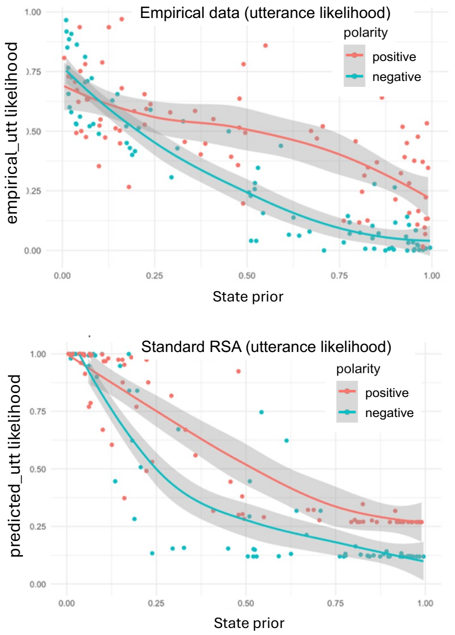


Figure 2: **a.** Empirically collected utterance likelihood (top) **b.** Model (standard RSA) predictions of utterance likelihood (bottom)

Figure 2a shows the utterance likelihoods from the human participants for both positive-polarity and negative statements. Visual inspection indicate that (i) for both sentence polarities, utterance likelihoods decrease as the state priors increase, (ii) for negative polarity, the decrease of utterance likelihoods as state priors increase is steeper than positive polarity. These patterns suggest a main effect of state prior and an interaction between state prior and sentence polarity.

Beta regression analysis confirms that there is a main effect of state prior (β = -4.36, SE = 0.28, z = -15.74, $p < .001$), and an interaction effect between state prior and sentence polarity (β = 2.43, SE =

0.35, z = 6.96, p < .001). From the positive sign of the interaction effect, we can confirm that the negative polarity yields a steeper decrease in utterance likelihood as the state prior increases. In addition, we found no main effect of sentence polarity (β = -0.20, SE = 0.20, z = -1.05, p = .296).

These results reveal patterns that Asymmetry Hypothesis 1 does not predict. On one hand, overall, positive utterances are not always perceived as having higher utterance likelihood. On the other hand, speakers are more likely to communicate low-informativeness information using positive polarity and more likely to communicate high-informativeness information using negative polarity.

Model predictions (standard RSA): Now let us see whether the standard RSA model can capture these observations. The model (as in (1)-(4)) is run in R using the rwebppl package[1].

The model considers two states: U_{state} = {s_{pos}, s_{neg}} and three possible utterances: $U_{utterance}$ = {u_{pos}, u_{neg}, u_{null}}. These utterances are mapped to truth values of different states. When the null utterance, u_{null} (say nothing), is made, people simply rely on their prior expectations (state prior) to interpret the situation. The positive utterance, u_{pos} "A has B", maps to the truth of only the positive state, s_{pos}. The negative utterance, u_{neg} "A doesn't have B", maps to the truth of only the negative state, s_{neg}.

The utterance utility term consists of an informativeness component, a cost component, and a speaker rationality parameter. α is set to 1 and utterance cost is specific to each of the three utterances (Cost(u_{null})=0; Cost(u_{pos})=1; Cost(u_{neg})= 2). *P(s)* is the normed state priors data that we plugged in the model as input.

Figure 2b shows the model-predicted utterance likelihoods for both sentence polarities. Visual inspection indicates that (i) similar to the empirical data, for both sentence polarities, utterance likelihoods decreased as the state priors increase, and (ii) for positive polarity, the predicted utterance likelihood is always higher than the negative. These patterns suggest a main effect of sentence polarity and a main effect of state prior.

Beta regression analysis reveals a main effect of state prior (β = -4.43, SE = 0.36, z = -12.47, p < .001). However, unlike human data, we found in the model predictions a main effect of sentence polarity (β = 0.78, SE = 0.28, z = 2.83, p < .01), indicating that the positive polarity always yields higher utterance likelihood than the negative polarity. Moreover, we did not find a significant interaction between state prior and sentence polarity (β = 0.13, SE = 0.44, z = 0.28, p = .78).

The results suggest that the standard RSA model follows predictions of the Hypothesis 1 and fails to fully capture the empirically observed patterns.

Comparing empirical data and model predictions: The discrepancy centers on the lower bound of the state prior that approaches a probability of 0: Based on human data, negative-polarity situations that have low priors (e.g., *The classroom doesn't have a board.*) are more likely to be communicated than positive-polarity situations that have similarly low priors (e.g., *The classroom has a stove.*). However, given that our human data were not collected in a spontaneous production study, it is possible that the Experiment 1 participants did not consider the role of utterance cost. We want to be cautious about committing to this pattern of sentence polarity asymmetry, so we ran another model simulation with the utterance cost constant as 1 for both sentence polarities.

Beta regression analysis now shows a main effect of state prior (β = -4.45, SE = 0.32, z = -13.75, p < .001), no effect of sentence polarity (β = 0.14, SE = 0.27, z = 0.53, p = .59), and no interaction between state prior and sentence polarity (β = 0.01, SE = 0.41, z = 0.02, p = .98). This shows that the model-predicted utterance likelihood of negative and positive sentences patterns alike, which is not surprising given how the model parameters do not differentiate them.

The results above suggest that even when cost is controlled, the standard RSA model fails to capture the sentence-polarity asymmetry observed in our empirical utterance likelihood data.

In the other model implementations in this paper, we thus assume higher cost for negative utterances than positive ones (also in line with cognitive psychology and linguistics research).

3.3 Common ground update

Experiment 2 (n=52) collected typicality ratings of the whole entity (e.g., *house*) using the same stimuli as in Experiment 1, except that the fact statement of a positive/negative part-whole relation was embedded in direct speech in Experiment 2, e.g., *"The house has a bathroom," Emma told her*

[1] https://github.com/mhtess/rwebppl

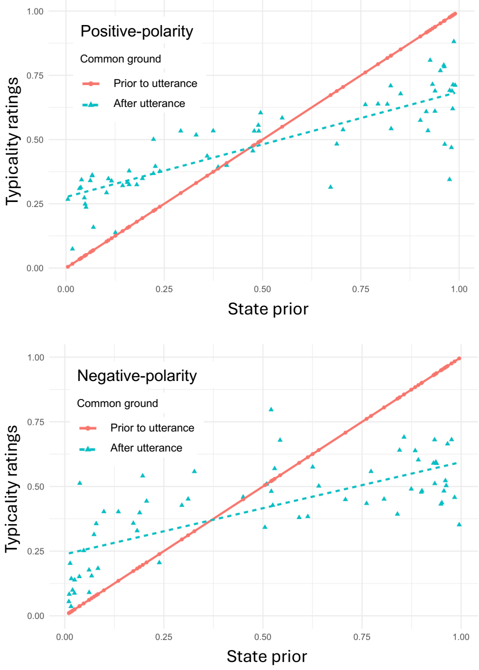


Figure 3: **a.** Ratings pre- vs. post- positive polarity utterances (top) **b.** Ratings pre- vs. post- negative polarity utterances (bottom)

partner. Participants were instructed to rate the typicality of *house* based on what the protagonist said about it (e.g., *How likely do you think it is that the house is a typical house?*).

Following standard RSA practice (Frank and Goodman, 2012; Degen et al., 2015), we compare ratings before and after utterances are presented to participants. Pre-utterance ratings (the norming data in Section 3.1) reflect (the listener's belief of) common ground prior to communication, while post-utterance typicality ratings (this section, Experiment 2) reflect updated common ground triggered by the utterance, in line with the discussions about Asymmetry Hypothesis 2.

Figure 3a shows these two types of ratings for positive polarity (solid line: state prior; dashed line: updated common ground). Figure 3b shows the same results for negative polarity.

The ratings were analyzed with Pearson correlation and beta regression. *First*, we assessed the correlation between state prior (norming) and typicality ratings (Experiment 2). To test this, we conducted a Pearson correlation: the typicality ratings are more strongly correlated with the state prior in positive polarity ($r_{utt}(76)$= 0.84, p<0.01) than in negative polarity ($r_{utt}(76)$= 0.75, p<0.01).

Second, to compare sentence polarities directly, we analyzed the interaction between polarity and state prior on typicality ratings using beta regression. We found a main effect of polarity (β= -0.41, SE = 0.08, z = -5.16, p<0.01) where the negative polarity yielded lower typicality ratings than the positive polarity, a main effect of state prior (β= 1.66, SE = 0.11, z = 15.01, p<0.01) where typicality ratings increased with state priors, but no interaction (β= 0.05, SE=0.22, z=0.21, p=0.83).

These results suggest that negation triggers stronger common ground update/inferences. However, importantly, our results suggest that (i) the positive-polarity is not free of inferences, and (ii) both sentence polarities can trigger atypicality inferences (Kravtchenko and Demberg, 2022ab) and what we call *typicality inferences* (i.e., low prior states are inferred to be more typical post- vs. pre-utterances).

Model predictions (standard RSA): The standard RSA model uses Boolean semantics, so the model updates the state posterior to 1 based on the only state that a non-null utterance makes true, but makes no inferences about common ground.

Comparing empirical data and model predictions: The comparison is fairly straightforward: The standard RSA model cannot handle common ground update.

Motivated by the discrepancies between empirical observations and model predictions (of the standard RSA), in the following Sections 4 to 6, we extend the standard model to better capture our empirical findings.

4 *fuzzy*RSA

The goal of Section 4 is to pinpoint the sentence polarity asymmetry related to the informativeness-cost tradeoff (i.e., a pragmatic speaker aims to maximize informativeness and minimize cost). Building on prior work, we introduce soft semantics into the standard RSA model to capture the asymmetry observed in utterance likelihood. We call this extended model the ***fuzzy*RSA model**.

4.1 Model

The *fuzzy*RSA model is extended from the standard RSA model by configuring different interpretation functions across sentence polarities. For a negative utterance, the fuzzy interpretation is defined as a constant probability distribution of a negative state and a positive one (see (11), where $n \in [0,1]$), with its optimal value determined during model fitting. For instance, when n is

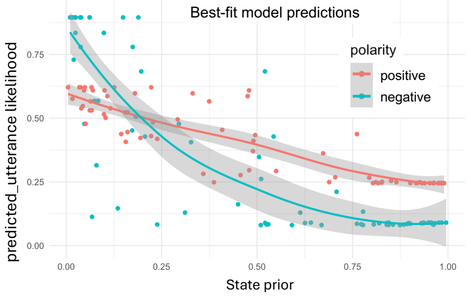


Figure 4: ***fuzzy*RSA.** Utterance likelihood values generated from the best-fit model.

assigned a value of .7, “A doesn’t have B” assigns a probability of .7 to “*A-no B*” and .3 to “*A-B*”.

$$[\![u_{neg}]\!](s_{neg}) = n,\ [\![u_{neg}]\!](s_{pos}) = 1 - n \quad (11)$$

This constant formulation reflects the ‘inherent’ pragmatic feature of negation as a presupposition trigger, which applies to all negative utterances.

For a positive utterance, the fuzzy interpretation is defined as a parametrized sigmoid function of the priors of positive states (see (12-13)), also fit during model optimization.

$$\begin{cases} [\![u_{pos}]\!](s_{pos}) = Sigmoid(P(s_{pos});\ \theta) \\ [\![u_{pos}]\!](s_{neg}) = 1 - [\![u_{pos}]\!](s_{pos}) \end{cases} \quad (12)$$

$$S_{\theta=\{L,k,x_0,c\}}\left(P(s_{pos})\right) = \frac{L}{1+e^{-k(P(s_{pos})-x_0)}} + c \quad (13)$$

The sigmoid function in (13) increases rapidly for state priors that are relatively low and gradually approaches the maximum value (i.e., approaching 1) towards relatively high state priors. The sigmoid function captures a systematic relationship between the state prior and the probability of interpreting a positive utterance as intended. Compared to the negative polarity, the interpretation function associated with positive polarity disincentivizes the communication of low-prior positive states.

4.2 Model fitting

We optimized model parameters by minimizing the joint loss across negative and positive polarities. This joint loss was computed as the sum of squared differences between model predictions and empirical data. A grid search over pre-specified parameter ranges—informed by exploratory model simulations—was used to identify the best fitting-values: n=.8, α=1, θ={L=0.7, k=6, x_0=.35, c=0.3}. The best-fit model has a mean square error (MSE) of 0.04 (compared to a MSE of 0.06 for standard RSA model).

4.3 Model predictions

Figure 4 shows that the *fuzzy*RSA model predicts patterns that resemble the empirical data. The results suggest that the *fuzzy*RSA model provides a better approximation of the empirical data and potentially of the cognitive processes involved in inferring utterance likelihood.

5 *wonky*RSA

In another extended model, we introduce a complex prior to capture the asymmetry in typicality ratings and provide a mechanism for common ground update. We call it the ***wonky*RSA model**.

5.1 Model

As discussed earlier, we integrate Cremers et al. (2023)’s modification into Degen et al’s (2015) ‘wonky world’ model, resulting in the following:

$$P_{L0}(s|u,w) \propto [\![u]\!](s) \cdot P(s|w) \quad (14)$$

$$P_{S1}(u|s,w) \propto exp(\alpha\ (ln\, P_{L0}(s|u,w) - Cost(u)) \quad (15)$$

$$P_{L1}(s,w|u) \propto P_{S1}(u|s,w)\ \cdot P(s|normal) \cdot P(w) \quad (16)$$

In the *wonky*RSA model, presupposition accommodation is reflected in an updated wonkiness, i.e., the wonky world has a higher or lower probability based on how much accommodation is needed.

Before the accommodation, the common ground is *P(s|w = normal)*. After the accommodation, the common ground is a complex probability distribution: *P(s|w = normal)* with a probability of (1-*P(w)*) and *P(s|w= wonky)* with a probability of *P(w)*. In other words, the updated common ground can be represented by the marginalized probability of a state across both worlds. We assume that the post-utterance ratings collected (typicality ratings; Experiment 2) reflect this updated common ground, which we refer to as *expected typicality*, formalized as following:

$$\mathbb{E}(typicality) = \textstyle\sum_{world} P(world) * P(s|world) \quad (17)$$

5.2 Model fitting

We optimized model parameters by minimizing the joint loss across negative and positive polarities. This joint loss was computed as the sum of squared differences between expected typicality and typicality ratings. A grid search over pre-specified parameter ranges—informed by exploratory model

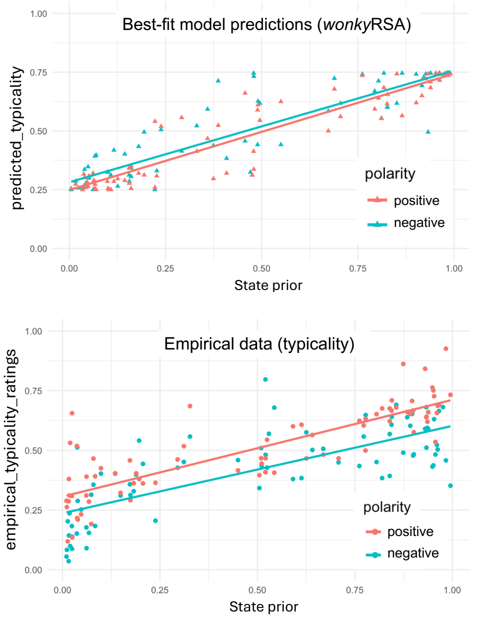


Figure 5: **a.** Model predictions of post-utterance expected typicality in positive vs. negative polarities (top) **b.** Post-utterance typicality ratings in positive vs. negative polarities (bottom)

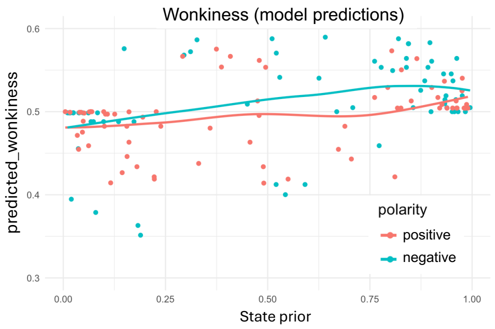


Figure 6: *wonky*RSA's predictions of **wonkiness**

simulations—was used to identify the best fitting-values: *w*=.5, *α*=2. The best-fit model has a MSE of 0.02 (while the standard RSA model is unable to make predictions regarding presupposition accommodation).

5.3 Model predictions

The best-fit model is able to capture two aspects of our empirical patterns. (i) Figure 5a shows that the *wonky*RSA model predicts both typicality and atypicality inferences in both sentence polarities. (ii) Figure 6 shows that the model-predicted wonkiness values more or less align with the inference patterns: lower-than-prior wonkiness is predicted where typicality inferences are observed, and higher-than-prior wonkiness is predicted where atypicality inferences are observed.

However, the *wonky*RSA model is not yet able to reflect the stronger inferences associated with the negative polarity: Instead of predicting lower typicality ratings for negative polarity (Figure 5b), the model predicts similar typicality values for both sentence polarities (Figure 5a).

This is not surprising given that the *wonky*RSA model does not differentiate two sentence polarities. Therefore, it is necessary to further extend the *wonky*RSA model, which we will discuss in Section 6.

6 *funky*RSA

In a third extended model, we bring together two approaches, soft semantics and the complex prior, from the preceding two models in Sections 4 and 5. This is an attempt to introduce polarity asymmetry into the *wonky*RSA model. We call this combinatory model the ***funky*RSA model**.

6.1 Model

The *funky*RSA model integrates components from *fuzzy*RSA and *wonky*RSA, formalized as shown:

$$P_{L0}(s|u,w) \propto [\![u]\!](s)\cdot P(s|w) \qquad (18)$$

$$\begin{cases} [\![u_{pos}]\!](s_{pos}) = Sigmoid(P(s_{pos});\ \theta) \\ [\![u_{pos}]\!](s_{neg}) = 1 - [\![u_{pos}]\!](s_{pos}) \end{cases} \qquad (19)$$

$$S_{\theta=\{L,k,x_0,c\}}\left(P(s_{pos})\right) = \frac{L}{1+e^{-k(P(s_{pos})-x_0)}} + c \qquad (20)$$

$$P_{S1}(u|s,w) \propto exp(\alpha\ (ln\, P_{L0}(s|u,w) - Cost(u)) \qquad (21)$$

$$P_{L1}(s,w|u) \propto P_{S1}(u|s,w)\ \cdot P(s|normal)\cdot P(w) \qquad (22)$$

6.2 Model predictions

Instead of fitting the model from scratch, we plugged in the values of parameters that contributed to the best-fit *fuzzy*RSA and *wonky*RSA models. Note that these two models differ in their values of the speaker rationality parameter α. We thus ran the *funky*RSA model with both values which yielded similar results for typicality. Figure 7 shows the model predictions of typicality in both polarities.

The model does predict a difference between sentence polarities; however, the predicted difference does not align well with the empirical findings: The negative polarity does not yield lower typicality values than the positive polarity.

This suggests that while optimal parameter values from *fuzzy*RSA and *wonky*RSA models

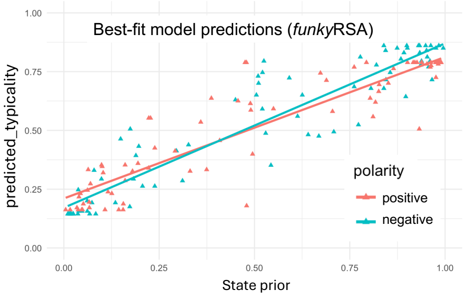


Figure 7: *funky*RSA model's predictions of typicality($\alpha = 1$).

provided a starting point, they do not yield satisfactory predictions when applied directly to the *funky*RSA model. Due to the increased complexity and computational cost of jointly optimizing all parameters in the *funky*RSA model, we leave full optimization for future work.

For utterance likelihood, we assume that the empirical ratings reflected participants' choices in a normal world. the *funky*RSA model makes the same predictions as the *fuzzy*RSA model regarding utterance likelihood.

7 Discussion

In this paper, we (i) empirically tested two hypotheses about sentence polarity asymmetries and (ii) introduced three extended RSA models that demonstrated the potential to better capture our empirical data than the standard RSA model.

The empirical data from Experiments 1 and 2 reveal patterns that are not predicted by the standard RSA model. Results of utterance likelihood ratings (Experiment 1) show that, although negation is theoretically deemed as a less optimal utterance choice than the positive polarity regarding the informativeness-cost tradeoff, negative utterances are not always less likely than positive utterances. Results of typicality ratings (Experiment 2) show that both state priors and sentence polarity play a role in triggering pragmatic inferences. Although negative utterances were associated with stronger inferences, positive utterances also yielded pragmatic accommodation.

To capture these novel empirical findings within the RSA framework, we targeted two components of an RSA model, namely the interpretation function that gives rise to literal meaning, and the configuration of common ground that allows presupposition accommodation. Inspired from prior work on soft semantics in RSA, our *fuzzy*RSA model uses different soft-semantics interpretation functions for different sentence polarities. Adapted from prior work on wonky world RSA models, our *wonky*RSA model provides a complex prior for common ground update. Combining *fuzzy*RSA and *wonky*RSA models, we then propose the *funky*RSA model which aims to introduce interpretation-level sentence polarity asymmetry into the *wonky*RSA model. The three extended RSA models yield somewhat better predictions than the standard RSA model and somewhat satisfying results that align better with the results of Experiments 1-2.

However, some questions remain open. *First*, regarding the different configurations in how different sentence polarities are literally interpreted, we formalized a sentence polarity asymmetry at a semantic level (i.e., through fuzzy interpretations). This worked for the predictions of utterance likelihood (*fuzzy*RSA model) but not for the predictions of typicality (*funky*RSA model), which might suggest that sentence polarity asymmetry is not limited to the difference in literal interpretations. Thus, future work should explore approaches to formalizing the sentence polarity asymmetry more closely related to common ground update. *Second*, regarding the complex prior used in the *wonky*RSA model, we explored one version of the wonky world—a uniform prior. This, however, is a potential source of sentence polarity asymmetry. For example, the wonky world assumed for negative utterances may differ from that for positive ones We plan to explore other configurations of the wonky world in future work.

8 Conclusion

This paper presents novel empirical findings on sentence polarity asymmetries and offers one of the first formalizations of these asymmetries within the RSA framework. The contributions are two-fold. Theoretically, this study highlights the important role of prior knowledge in pragmatic reasoning and offers new insights into both production and comprehension of negation. Empirically, we show that existing extensions of the RSA model, e.g., soft semantics and common ground update, while not originally proposed to address sentence polarity asymmetries, can nonetheless be applicable to these phenomena. This supports the generalizability of these approaches, as well as strengthens the broader applicability of the RSA framework.

Acknowledgments

We thank the anonymous reviewers for thoughtful feedback on this project.

References

Bergen, Leon, Noah Goodman, and Roger Levy. 2012. That's what she (could have) said: How alternative utterances affect language use. In Proceedings of the Annual Meeting of the Cognitive Science Society, 34(34).

Bergen, L., & Goodman, N. D. 2015. The strategic use of noise in pragmatic reasoning. *Topics in Cognitive Science*, 7(2), 336–350.

Cornish, E. R., & Wason, P. C. 1970. The recall of affirmative and negative sentences in an incidental learning task. *Quarterly Journal of Experimental Psychology*, 22(2), 109–114.

Cremers, A., Wilcox, E. G., & Spector, B. 2023. Exhaustivity and Anti-Exhaustivity in the RSA Framework: Testing the Effect of Prior Beliefs. *Cognitive Science*, 47(5), e13286.

Degen, Judith, Michael Henry Tessler, and Noah D. Goodman. 2015. Wonky worlds: Listeners revise world knowledge when utterances are odd. In Proceedings of the 37th Annual Meeting of the Cognitive Science Society, 548–553. Austin, TX: Cognitive Science Society.

Degen, J., Hawkins, R. D., Graf, C., Kreiss, E., & Goodman, N. D. 2020. When redundancy is useful: A Bayesian approach to "overinformative" referring expressions. *Psychological Review*, 127(4), 591–621.

Degen, J. 2023. The rational speech act framework. *Annual Review of Linguistics*, 9(1), 519–540.

Frank, Michael C., and Noah D. Goodman. 2012. Predicting pragmatic reasoning in language games. Science 336(6084):998. https://doi.org/10.1126/science.1218633

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

Givón, T. 1978. Negation in language: Pragmatics, function, ontology. In *Pragmatics*, 69–112. Brill.

Hayfield, Tristen, and Jeffrey S. Racine. 2008. Nonparametric econometrics: The np package. Journal of Statistical Software 27(5).

Horn, Laurence R. 1989. A Natural History of Negation. Chicago, IL: University of Chicago Press.

Jakobson, R. 1963. Implications of language universals for linguistics. In *Roman Jakobson: Selected Writings II*, 580–592. The Hague: Mouton.

Kaup, B., Lüdtke, J., & Zwaan, R. A. 2006. Processing negated sentences with contradictory predicates: Is a door that is not open mentally closed? *Journal of Pragmatics*, 38(7), 1033–1050.

Kravtchenko, Ekaterina, and Vera Demberg. 2022a. Informationally redundant utterances elicit pragmatic inferences. Cognition 225:105159. https://doi.org/10.1016/j.cognition.2022.105159

Kravtchenko, E., & Demberg, V. 2022b. Modeling atypicality inferences in pragmatic reasoning. In Proceedings of the Annual Meeting of the Cognitive Science Society, 44(44).

Maciuszek, J., & Polczyk, R. 2017. There was not, they did not: May negation cause the negated ideas to be remembered as existing? *PLoS One*, 12(4), e0176452.

Qing, Ciyang, Noah D. Goodman, and Daniel Lassiter. 2016. A rational speech-act model of projective content. In Proceedings of CogSci.

Roberts, C. 1996. Information structure: Towards an integrated formal theory of pragmatics. In Je Hak Yoon and A. Kathol (eds.), *Ohio State University Working Papers in Linguistics (OSUWPL)*, 49: Papers in Semantics. Columbus, OH: The Ohio State University Department of Linguistics.

Roberts, C. 2012. Information structure in discourse: Towards an integrated formal theory of pragmatics. *Semantics and Pragmatics*, 5, 1–69. [Reprint of Roberts (1996).]

Scontras, Gregory, Michael Henry Tessler, and Michael Franke. 2021. A practical introduction to the Rational Speech Act modeling framework. arXiv preprint arXiv:2105.09867.

Stalnaker, Robert. 1978. Assertion. In Formal Semantics: The Essential Readings, 147-161.

Stalnaker, Robert. 2002. Common ground. Linguistics and Philosophy 25(5/6):701-721. https://doi.org/10.1023/A:1020867916902

Warstadt, Alex. 2022. Presupposition triggering reflects pragmatic reasoning about utterance utility. In Proceedings of the 2022 Amsterdam Colloquium.

Yoon, Eunice J., Michael Henry Tessler, Noah D. Goodman, and Michael C. Frank. 2020. Polite speech emerges from competing social goals. Open Mind 4:71-87. https://doi.org/10.1162/opmi_a_00035

Zadeh, L. A. 1978. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1), 3–28.

Is analogy enough to draw novel adjective-noun inferences?

Hayley Ross,[1] Kathryn Davidson,[1] Najoung Kim[2]
[1]Harvard University [2]Boston University
hayleyross@g.harvard.edu kathryndavidson@fas.harvard.edu najoung@bu.edu

Abstract

Recent work (Ross et al., 2025, 2024) has argued that the ability of humans and LLMs respectively to generalize to novel adjective-noun combinations shows that they each have access to a compositional mechanism to determine the phrase's meaning and derive inferences. We study whether these inferences can instead be derived by analogy to known inferences, without need for composition. We investigate this by (1) building a model of analogical reasoning using similarity over lexical items, and (2) asking human participants to reason by analogy. While we find that this strategy works well for a large proportion of the dataset of Ross et al. (2025), there are novel combinations for which both humans and LLMs derive convergent inferences but which are not well handled by analogy. We thus conclude that the mechanism humans and LLMs use to generalize in these cases cannot be fully reduced to analogy, and likely involves composition.

1 Introduction

How are humans able to generalize to complex linguistic expressions they have not encountered before? One view on how this can be achieved is through a mechanism of composition, determining the meaning of the phrase and any resulting inferences from the meanings of its parts (Partee, 2009; Szabó, 2012, i.a.). Others, however, believe that composition is not required: mechanisms such as analogy are sufficient to explain humans' ability to generalize to novel phrases (Bybee, 2010; Ambridge, 2020 i.a.). The same question arises when we study LLMs' ability to generalize. If they can generalize to novel phrases, is this evidence that they must be composing these phrases from their subparts, or is there another way to achieve the same results?

Ross et al. (2025) argue that humans must be using composition, since they converge on the inferences of at least some combinations that they are assumed never to have seen before (e.g., for *fake reef* or *counterfeit scarf*, which never appear in a large corpus). Ross et al. (2024) suggest a similar conclusion for LLMs based on the same dataset, since LLMs show reasonably human-like behavior on at least some bigrams that are assumed not to be in the LLMs' training datasets. These combinations are interesting because the membership inferences targeted (e.g., "Is a counterfeit scarf still a scarf?") depend not just on the adjective but also on the noun, involving significant detail about how exactly the adjective affects the noun and what properties are important for membership in that noun category in typical situations.

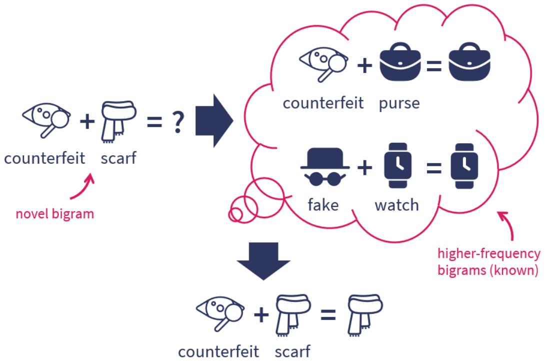


Figure 1: Possible analogical reasoning to infer that *counterfeit scarf* is a scarf, since a *counterfeit purse* is a purse and a *fake* (or *counterfeit*) *watch* is a watch.

This paper questions these conclusions, and investigates whether this task can in fact be solved by analogical reasoning, without composition. For example, for *counterfeit scarf*, one might reason (as in Figure 1): "Is a counterfeit scarf still a scarf? A scarf is an accessory like a watch or a purse, and a counterfeit watch is still a watch, and a counterfeit purse is still a purse, so a counterfeit scarf is most likely still a scarf". This skips the compositional step of combining the meanings of the words to derive the meaning of the bigram and further violates the principle of compositionality as stated by Szabó (2012) by referring to information beyond the meaning of the bigram's parts, namely the inferences associated with other adjective-noun bigrams.

We investigate analogical reasoning through two complementary approaches. First, we build a computational model of analogical reasoning which attempts to derive ratings for the low-frequency and zero-frequency (assumed novel) bigrams in the dataset of Ross et al. (2025), by analogy to the high-frequency ones. A computational model allows us to precisely define what we mean by analogy, and explore the consequences of different implementation decisions. Second, we ask human participants to reason analogically, guided by examples and their own intuition of what analogy means. We then evaluate how often they can produce an analogy, and whether the resulting rating distributions derived analogically are the same as the distributions from Ross et al. (2025), where no instructions on how to reason were given. We find that the ratings derived by analogy significantly differ for several bigrams, suggesting that the original participants did not derive (all) their ratings by analogy.

Between the two methods, we find convincing evidence that while analogical reasoning produces similar results in many cases, it is not sufficient to derive the full set of inference data. Thus, we find support for the view that humans must have access to a compositional mechanism. Further, our analogy model performs worse on novel bigrams than the best LLM in Ross et al. (2024), and our analogy model's successes and failures correlate poorly with those of the best LLM. This suggests that the LLM is not (just) using analogy in the cases where it can generalize, and supports the claim in Ross et al. (2024) that such LLMs are performing some kind of composition (productively combining the meaning of adjective and noun) in these cases. We share our code and data on GitHub.[1]

2 Related Work

So-called "privative" adjectives such as *fake* pose a challenge for compositional accounts of semantics, since they cannot be simply intersected with the noun (Kamp and Partee, 1995). Multiple accounts have been proposed for how composition with privative adjectives should work (Partee, 2010; del Pinal, 2015; Martin, 2022; Guerrini, 2024 i.a.).

Most previous computational work on adjective-noun composition using distributional semantics does not discuss privative adjectives (Baroni and Zamparelli, 2010; Vecchi et al., 2017; Hartung et al., 2017). Boleda et al. (2013) cover 16 "non-intensional" adjectives, including two which are commonly taken to be privative (*former*, *mock*; see Nayak et al. (2014) for a classification). Boleda et al. build distributional semantic models of adjective-noun composition that use vector addition and matrix multiplication to model adjective-noun composition, but they do not cover analogy. Cappelle et al. (2018) study the distributional semantics of *fake* and bigrams in which it occurs, but do not implement any method of composition or generalization.

Ross et al. (2025) gather a large quantity of offline human judgments on (privative) adjectives and their membership inferences, discussed further in Section 3, and Ross et al. (2024) extend this dataset to assess LLMs. While Ross et al. (2024) do propose a simple analogy baseline to compare to their LLMs, we propose an improved, more powerful and configurable analogy model and present a detailed analysis of its performance.

Analogy has been much studied as a core component of human reasoning (see Hofstadter, 2001 for an overview), and approaches such as construction grammar propose that analogy to known exemplars can be used to understand any novel phrase (Bybee, 2010; Ambridge, 2020). Rambelli et al. (2024) propose a computational model of this process based on distributional semantics. While we also build our computational model around analogy between phrases, we only attempt to derive membership inferences from the analogy, and avoid commitment to whether the full meaning of the phrase can be accessed by analogy.

3 Human Judgment Dataset

Ross et al. (2025) present a dataset of human judgments on adjective-noun inferences of the form "Is an {adjective} {noun} still a {noun}?" on a 5-point Likert scale. The dataset covers 798 bigrams (102 nouns crossed with 6 typically privative and 6 typically subsective adjectives, filtered to only include combinations that make sense).[2] In this dataset, the

[1] https://github.com/rossh2/artificial-intelligence/

[2] In this paper, we follow Ross et al. (2025) in using "(typically-) privative / subsective adjective" to refer to adjectives historically classified as such, which often but not always result in the respective inference.

question is presented out of the blue as a generic, rather than in a discourse context. The additional information in a discourse can sometimes determine the inference on its own (without needing to interpret the bigram at all), whereas the out of the blue setting requires some kind of reasoning strategy (composition, analogy or otherwise) to determine the inference. 180 of the 798 bigrams are zero frequency in the C4 pretraining corpus (Raffel et al., 2020), which Ross et al. (2024) take as a proxy for the undisclosed pretraining corpora of the models they study. These bigrams are assumed to be novel to both humans and LLMs. A bigram is referred to as high frequency if it is in the top quartile of bigrams they study.

Ross et al. (2025) show that the membership inference in question depends on both the adjective and the noun, with bigrams with "subsective" adjectives usually yielding subsective inferences (e.g., "a homemade N is an N", but not always: consider *homemade cat*), while bigrams with "privative" adjectives such as *fake crowd* elicit a wide distribution of ratings from subsective ("is") to privative ("is not"), with high variance for many (but not all) bigrams. Varying ratings between participants are expected in this setting, since we are dealing not only with the lexicon but also with a broad question (a linguistic generic) which may depend on participants' world knowledge. Participants nonetheless show convergent ratings for many zero-frequency bigrams, demonstrating their ability to generalize and implying a shared underlying mechanism.

4 Analogy Model

4.1 Algorithm

We implement a computational model of analogy which is "trained" on the human ratings from Ross et al. (2025) for a set of common (high-frequency) bigrams, which are stored in the model's memory. This is intended to imitate human prior experience with certain bigrams, where they may have learned that, for instance, a *counterfeit watch* is still a *watch*. Humans are known to store frequent multi-word expressions even when those expressions are compositional, not just when they are idiomatic (Arnon and Snider, 2010; Tremblay and Baayen, 2010; Caldwell-Harris et al., 2012, i.a.), so it is plausible to assume that they can also store the associated inferences. Specifically, we consider the top quartile of bigrams in Ross et al. (2025) as "known", i.e., in the training set. (Appendix C also explores an alternative approach where the training set is balanced evenly across adjectives.)

Given these known bigrams, the model predicts the ratings for the remainder of the bigrams by analogy to similar bigrams in its training set, via the algorithm in Figure 2. The setting `mem` configures whether this algorithm is also applied to bigrams in the training set, as if they were not known; we discuss in Section 4.4 what is more human-like.

The model stores and predicts the entire rating distribution for each bigram, rather than a single rating. As Ross et al. (2024) discuss in the context of LLMs, it is not clear how to evaluate the alignment of a single rating against high variance distributions like the human data we are taking as the evaluation target. As discussed in Section 3, such high variation is a natural consequence of working with the lexicon, but does necessitate a more complex metric than just accuracy to assess model fit. We use same metric that Ross et al. (2024) use for LLMs: the Jensen-Shannon divergence between the model-predicted rating distribution and the human rating distribution for each bigram. We compute an aggregate score by averaging across all bigrams. We report this aggregate score as well as the average score over zero-frequency bigrams (presumed to be novel to both humans and LLMs) to measure its ability to generalize. These zero-frequency bigrams are always held out from the model.

Implementing analogical reasoning in a computational model allows us to define precisely what we mean by analogy and test the effects of these implementation choices. We explore two types of analogy: either just over nouns (*counterfeit scarf* → *counterfeit watch*),[3] or allowing analogy over both noun and up to one additional adjective (*counterfeit scarf* → *fake watch*; `N+A` setting). We allow the model to retain $k \leq 5$ nearby bigrams (after filtering to bigrams in the training set) to impose constraints akin to human working memory (Cowan, 2001; Adam et al., 2017). The exact value of k is a hyperparameter optimized on the training set (with memorization disabled). Appendix C also discusses the case where $k = 1$, i.e. where the model only considers the most similar bigram, which is a plausible route for humans.

We calculate word similarity in three ways: (1) cosine similarity over GloVe embeddings (Pennington et al., 2014); (2) cosine similarity over

[3] We see in Section 5 that this is a popular human strategy: humans choose an analogy over just nouns 58% of the time.

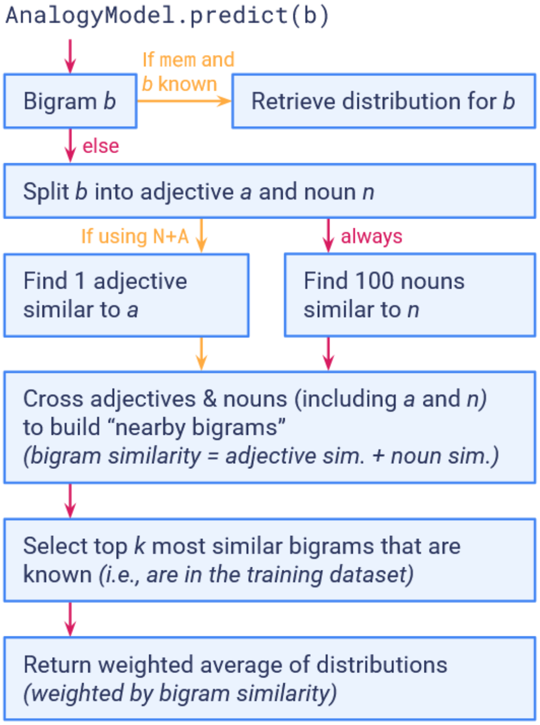

AnalogyModel.predict(b)
If mem and b known
Bigram *b*
Retrieve distribution for *b*
else
Split *b* into adjective *a* and noun *n*
If using N+A
always
Find 1 adjective similar to *a*
Find 100 nouns similar to *n*
Cross adjectives & nouns (including *a* and *n*) to build "nearby bigrams"
(bigram similarity = adjective sim. + noun sim.)
Select top *k* most similar bigrams that are known *(i.e., are in the training dataset)*
Return weighted average of distributions
(weighted by bigram similarity)

Figure 2: Algorithm for the analogy model. Yellow paths are dependent on the configuration options `mem` and `N+A` (Noun + Adjective). k is a hyperparameter.

embeddings from Llama 3 70B Instruct (Dubey et al., 2024) and (3) Wu-Palmer similarity over the WordNet taxonomy (Wu and Palmer, 1994; Miller, 1995). Llama 3 70B Instruct was selected as the source for LLM embeddings because this was the model with the highest performance in Ross et al. (2024). To derive word embeddings from Llama, we pass each word individually to the LLM and average the hidden states of the subword tokens in the final layer.[4] Wu-Palmer similarity groups nouns[5] that share common hypernyms in WordNet, penalized by how broad that hypernym is. Using WordNet allows us to measure similarity based solely on a human-created dataset, as opposed to distributionally derived embeddings. Since WordNet does not provide a taxonomy of adjectives, this approach is limited to noun-only analogies.

4.2 Results

Figure 3 shows the performance of the different analogy model configurations on the whole dataset (allowing memorization of the training set) and on held-out, zero-frequency bigrams (assumed to be novel to humans and LLMs). More details, including results for privative adjectives only and for single-bigram analogies ($k = 1$), are given in Appendix C (Table 1).

GloVe embeddings. Both the noun-only and N+A setting perform well overall, with the N+A setting appearing to be on par with LLM performance. However, we find that this is reliant on memorizing the training set; neither setting generalizes well to zero-frequency bigrams. In particular, noun-only analogies perform below a uniform distribution baseline on zero-frequency bigrams.

WordNet. Perhaps surprisingly, we find that this qualitatively different similarity metric yields very similar results to using GloVe embeddings, at least in the noun-only case where this metric is defined. We discuss the implication further in Section 4.3.

Llama Embeddings. Using the embeddings derived from Llama 3 70B Instruct also does not improve performance significantly compared to using GloVe, though we see a small increase for the noun-only setting—see also the discussion in Section 4.3.

Error Analysis. To investigate where the analogy model fails, we fit a linear regression in R (R Core Team, 2023) that predicts the JS divergence of the best-performing model from the adjective class (subsective vs. privative), human rating mean and human rating SD, with an interaction between adjective class and mean. Including the human SD allows us to target bigrams with divergent ratings; including an interaction of adjective class and mean allows us to pick out e.g. bigrams with subsective adjectives but privative ratings.

All main effects and the interaction are significant: JS divergence is lower for privative-class adjectives, higher for bigrams with subsective-class adjectives with privative ratings (i.e., low mean ratings, such as *homemade money* or *tiny abundance*), higher for privative-class bigrams with subsective ratings (i.e., high mean ratings, such as *false rumor* or *counterfeit watch*), and lower for bigrams with a high human standard deviation. The fact that it struggles on bigrams like *homemade money* (JS $= 0.81$) and *tiny abundance* (JS $= 0.58$) in

[4] We could alternatively pool the embeddings from the initial embedding layer, but the absence of contextualization in this approach may degrade results for multi-token words (~40% of our dataset). Nevertheless, we show in Appendix C that results are similar in this setting.

[5] Strictly, the metric groups noun synsets ("senses"); we use the 2 most common synsets per noun.

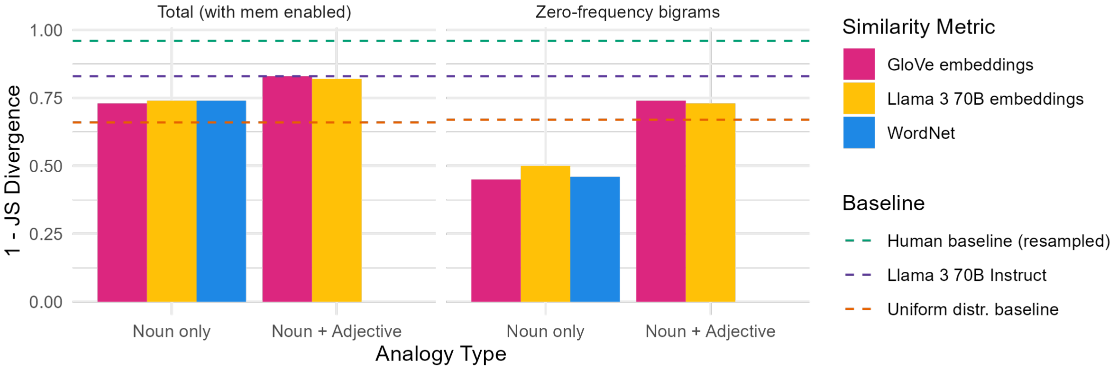


Figure 3: Average JS divergence between distributions produced by the analogy model and human distributions from Ross et al. (2025) on zero-frequency bigrams and on the whole dataset (with memorization of the training set). Additional results are given in Table 1 in the Appendix.

particular is not surprising, given that these adjectives are subsective for all except two bigrams in the model's pool of analogy candidates.

4.3 Discussion: Effect of Similarity Metric

The similarity metric used is not a main modulator of model performance. One possible explanation is that the analogies found by our model may often be suboptimal or inadequate, regardless of the similarity metric used. There are two potential sources of this inadequacy: first, analogical reasoning may inherently be a flawed approach for some bigrams. Second, the training set may be so sparse that the model cannot retrieve sufficiently similar nouns or bigrams to adequately support analogical reasoning. After all, our training set contains ratings for only 279 bigrams using 89 nouns (of 102 nouns in the original dataset).[6] While we cannot fully tease these two possibilities apart with our current experiments, Appendix E explores adding data from the human rating experiment in Section 5.

4.4 Discussion: Humans

Working with lexical semantics requires us to deal with per-bigram distributions and a distribution comparison metric, rather than proportions of correct answers or significant effects in a regression. This makes interpretation of the results more complicated. It is not clear at what threshold to conclude that the model captures human performance, versus what amount of JS divergence represents noise/artifacts generated by the relatively small distribution sample size in the human experiment ($n = 12$ per bigram). Short of replicating the human experiment in Ross et al. (2025) and calculating the JS divergence between the two, we have three points of reference: (1) We can approximate a human JS divergence by resampling from the human distribution. This yields an average JS divergence of just 0.05; (2) The best LLM performance that achieves JS divergence of 0.17 both overall and on zero-frequency bigrams (Ross et al., 2024); (3) The ratings collected from the experiment in Section 5, where humans are asked to perform the same task as the analogy model, yield an overall JS divergence of 0.16 compared to the original distributions.

Our analogy model achieves a JS divergence of 0.17 at best, when allowed to memorize its training data; 0.25 when it does not memorize it. On zero-frequency bigrams, the best score is 0.25. While the results are impressive with memorization, its ability to generalize to zero-frequency bigrams is 8 points worse than LLMs and 11 points worse than humans. This suggests that our analogy model does not fully capture human behavior. While a key part of the modeling assumption is that the training data represents humans' known and memorized bigrams, it is still unclear whether it is human-like to return the exact perfect distribution—all the more so considering that we typically ask humans to give single ratings, not entire distributions.

As an alternative metric, we conduct per-bigram Kolmogorov-Smirnoff tests (Holm-Bonferroni adjusted) comparing the distributions predicted by the analogy model to the human distributions. We find that with memorization of the training set, 10 of the predicted distributions are significantly dif-

[6] The 102 nouns were selected by Ross et al. such that each noun has at least one closely related other noun.

ferent ($p < 0.05$), of which 3 are zero-frequency bigrams; without memorization, this rises to 20. Since we only have a sample size of $n = 12$, this is a conservative estimate. Figure 8 in Appendix C.3 shows a selection of such distributions. The fact that the analogy model significantly deviates from the correct distribution for these cases supports our conclusion that while analogy is successful in most cases, it does not offer a full explanation.

4.5 Discussion: LLMs

It may seem striking that the analogy model can achieve the same overall JS divergence as Llama 3 70B Instruct, the best model studied by Ross et al. (2024), when we allow training set memorization. However, comparing results on the zero-frequency bigrams (and also on performance without `mem`, see Table 1) shows that Llama 3 70B Instruct generalizes much better than our analogy model. Further, fitting a linear regression to predict the LLM's JS divergence per-bigram from the Llama embedding analogy model's divergence shows that although the effect is significant ($p < 0.001$), this only explains 12% of the variance in the LLM's ratings ($R^2 = 0.12$; $R^2 = 0.04$ with `mem` enabled). In other words, the LLM's behavior is not particularly well explained by the analogy model, and it does not succeed and fail in the same places.

5 Human Analogical Reasoning

While the analogy model allows us to precisely control the mechanism and data used for analogical reasoning, it also suffers from an artificial restriction on the bigrams to which it can draw an analogy: its training dataset is strictly limited to the bigrams that Ross et al. (2025) gathered human ratings for. Actual human analogical reasoning would not be limited in the same way, and is likely to involve a much wider range of analogy targets. In this experiment on human participants, we expand the definition of analogy to whatever our participants construe as analogy (given our instructions and training examples), enabling access to whatever bigrams they are able to come up with as suitable analogies. This allows us to measure two things: (1) how easy it is for people to come up with analogies at all, and (2) what effect analogical reasoning has on the resulting rating distributions.

5.1 Method

We select 96 bigrams from the 798 bigrams from Ross et al. (2025) such that they are evenly balanced by adjective and by zero vs. top quartile frequency, and all have convergent human rating distributions ($\mu \leq 2$ or $\mu \geq 4$ on the 5-point scale).[7]

For each bigram, we show participants the question "Is an {adjective} {noun} still a {noun}?" and first ask them whether they are able to come up an analogy that helps them answer the question. We then ask them to answer the question, either using the analogy or not, depending on their first answer. Screenshots of each path are shown in Figure 4. Participants first see an explanation of what we mean by analogy, including an example (*toy hippo* → *toy elephant*), followed by three training examples which include another example of an analogy (*melted plastic* → *melted wax/chocolate*). The full instructions, including our description of "analogy", are given in Appendix F. The analogy text field is limited to 1-3 words to encourage analogy to adjective-noun phrases (pilot participants sometimes typed a reasoning process into the field).

We recruited 176 native American English speakers[8] on Prolific, of which we excluded 33 for not meeting our native speaker criteria, failed attention checks, or failing to adequately follow our instructions for analogical reasoning (verified based on manual inspection and regular expression searches on the free text entry fields).

5.2 Results

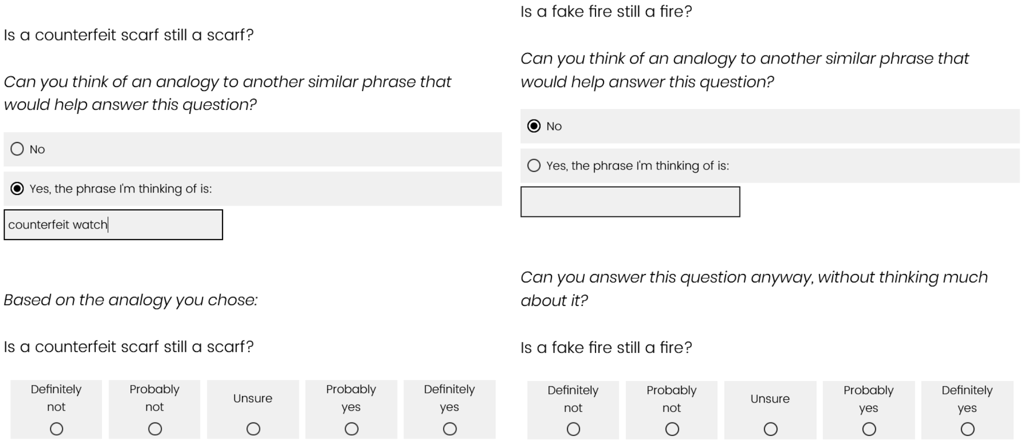
Overall, participants self-reported that they could find an analogy for 56.4% of responses. For every bigram except *fake impression*, at least one person was able to find an analogy, although 13 of 143 participants never produced an analogy. A plot of analogy availability for each bigram is shown in Figure 7 in Appendix A.[9]

Type of analogy. Figure 5 shows statistics for the types of analogy drawn. We find that 58.4% of analogies use the same adjective as the original bigram, such as *knockoff watch* → *knockoff purse*, while only 10% change the adjective and use the same noun, such as *homemade money* → *counterfeit money*. A further 6.2% of analogies use a single noun. While a number of these single-noun analogies seem intended as same-adjective analo-

[7] We also attempt to include a high proportion of bigrams where analogy might be hard—see Appendix D. For example, we adversarially pick some nouns for *homemade* which are likely to yield privative judgments, such as *homemade money*.

[8] See Appendix B for detailed criteria.

[9] We attempted a regression to predict analogy availability but found nothing of interest; see Appendix D.



Is a counterfeit scarf still a scarf?

Can you think of an analogy to another similar phrase that would help answer this question?

○ No

● Yes, the phrase I'm thinking of is:

counterfeit watch

Based on the analogy you chose:

Is a counterfeit scarf still a scarf?

Definitely not	Probably not	Unsure	Probably yes	Definitely yes
○	○	○	○	○

(a) Path when analogy found.

Is a fake fire still a fire?

Can you think of an analogy to another similar phrase that would help answer this question?

● No

○ Yes, the phrase I'm thinking of is:

Can you answer this question anyway, without thinking much about it?

Is a fake fire still a fire?

Definitely not	Probably not	Unsure	Probably yes	Definitely yes
○	○	○	○	○

(b) Path when no analogy found.

Figure 4: Screenshots of questions in the analogy prompting experiment.

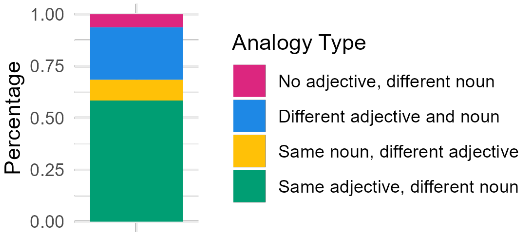


Figure 5: Types of analogy chosen by participants.

gies (such as *tiny bed* → *(tiny) chair*), we do see some interesting cases such as *artificial rumor* → *lie*, which may not be an analogy in the strict sense but are still solving the task by mapping to a known phrase. The remaining 25.4% use a different adjective/modifier and noun.

Qualitatively, we see that our participants reach for a much wider set of concepts than our analogy model when drawing analogies; choices such as *homemade lake* → *homemade cookies*, *false impression* → *wrong interpretation* or even *multicolored weapon* → *painted nails* are common. Participants are more likely than our model to reach for nouns that are not that similar to the original noun but are highly associated with the adjective, such as *knockoff purse* (11 occurrences as analogy), *counterfeit money* (10 occurrences), *homemade cookies* or *illegal immigrant* (3 occurrences each).

Distribution shift. Does analogical reasoning shift the distribution compared to the original ratings gathered by Ross et al. (2025), where no instructions on how to reason were provided? In the cases where an analogy was found, we find an average JS divergence of 0.16 overall between bigram distributions in this experiment vs. in Ross et al. (2025), with 0.21 on privative-type adjectives (0.32 for *fake*), 0.35 on *homemade* (recall that nouns for *homemade* were picked adversarially to be more likely to be privative) and 0.14 on zero-frequency (presumed novel) bigrams.

We also conduct Kolmogorov-Smirnoff tests per-bigram (with Holm-Bonferroni adjustment) to determine which of the distributions are significantly different. Since our n per bigram is quite small for statistical purposes (at best $n = 12$, lower if not all participants found an analogy for the bigram), no bigrams are significantly different. We cannot conclude from this that the distributions are indeed the same when analogy is used; the sample size is just too limited. Instead, we plot the distributions for 6 bigrams with the highest JS divergences in Figure 6. The divergence for *homemade currency* and *homemade money* (and to a lesser extent *false friend*) is particularly striking: analogy leads people to dramatically different inferences in these cases, since most *homemade* and many *false* items (such as *false rumor*) still clearly qualify as an instance of the noun.

Correlation between analogy availability and distribution shift. We fit a beta regression in R (Brooks et al., 2017) that predicts JS divergence as a function of analogy availability. We find a strong negative correlation: JS divergence decreases as analogy availability increases ($p < 0.001$). In other words, the harder it is to find an analogy, the more likely any analogies that are found will lead people astray from the original distribution.

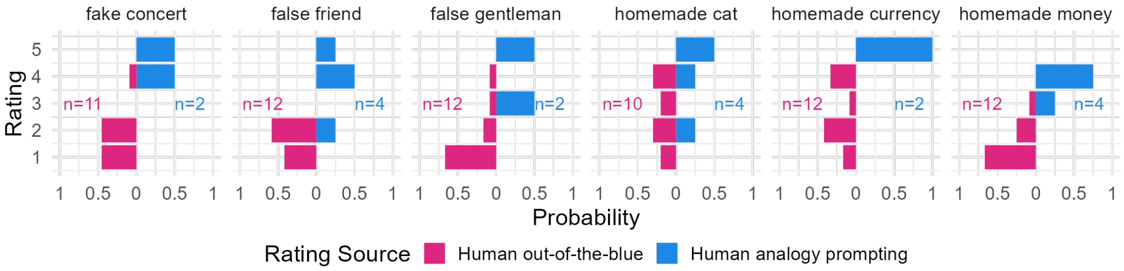


Figure 6: Distributions for the 6 bigrams with the highest JS divergences when an analogy is used. n = number of ratings in each distribution; for analogy prompting, this is however many people found an analogy.

5.3 Discussion

This experiment shows that analogy is a viable approach for many bigrams, and in many cases results in similar judgments as in Ross et al. (2025), where participants could reason freely. However, for several bigrams such as *homemade money*, using an analogy yields dramatically different inferences, suggesting that analogy was not used to derive the original distribution. We also see bigrams where people struggle to come up with any analogy at all, such as *fake impression* ($n = 0$). This was the case for 10 of our 35 zero-frequency bigrams ($n \leq 50\%$), putting into question the viability of analogical reasoning for generalization. Our analogy model also shows a higher-than-average JS divergence for all bigrams (except one) where analogical reasoning substantially shifts human ratings. It also shows a higher-than average JS divergence for over half the bigrams where humans struggle to come up with an analogy. Overall, a linear regression predicting human JS divergence from the analogy model's JS divergence explains 40% of variation, suggesting that analogy serves as a viable explanation for some, but not all of the variation in human inferences. As for LLM behavior, human analogy availability and human-human JS divergence when using analogies both correlate poorly with LLM-human JS divergence per-bigram, with $R^2 = 0.05$ in both cases ($p = 0.03$ and $p = 0.04$ respectively). A similar regression with our analogy model in Section 4.5 also showed low correlation. This suggests that analogical reasoning poorly explains LLM behavior, corroborating our previous conclusion in Section 4.5.

Finally, we observe that our participants use a much broader definition of "analogy" than our analogy model (or the examples we gave during training), suggesting that our model adheres to adjective and noun similarity overly strictly. Further, our analogy model is strictly non-compositional at the meaning level, whereas some human analogies such as *false impression* → *wrong interpretation* may well be arising from the participants first composing the meaning of *false impression* and then looking for phrases with a similar meaning.[10]

6 Conclusion

Ross et al. (2025) claim that humans must be handling adjective-noun bigrams compositionally, since they draw consistent inferences about novel bigrams, and Ross et al. (2024) take LLMs' capacity to draw reasonably human-like inferences on the same novel bigrams as evidence for composition. We explored the possibility that this generalization might be explained without composition in either or both cases, specifically by analogical reasoning over adjective and nouns using previously encountered and memorized inferences.

Composition in humans. We find that while many of the novel bigrams in the dataset can indeed be handled successfully by analogy, analogy is not sufficient to explain human behavior fully. Our analogy model diverges significantly from human distributions on 20 bigrams and shows insufficient generalization to zero-frequency bigrams, with a JS divergence of 0.25 from humans. Humans both struggle to come up with analogies for 24% of bigrams tested and are led astray when they do for several bigrams, such as *homemade currency*. We thus conclude that analogical reasoning is a successful strategy for generalization in a remarkable proportion of the dataset of Ross et al. (2025), but analogy does not suffice to handle the full data. Thus, their conclusion that some

[10] *False* may mean *not truthful/insincere* or just *fake* (as in *false teeth*); the choice of meaning depends on the noun.

mechanism of composition seems necessary to handle the whole range, *homemade currency* and all, is supported—even if humans need not (and judging by our data, quite possibly do not) invoke it in every case. This conclusion is similar to the result of Albright and Hayes (2003), who found that an analogical model of English past tense morphology did not explain participant behavior well, and concluded that speakers used abstract rules to generalize rather than analogy.

Composition in LLMs. We likewise find that LLM behavior can be partially, but not fully explained by analogical reasoning. Our analogy model is unable to reach the performance of the most successful LLMs in Ross et al. (2024), in particular when generalizing to zero-frequency bigrams. Moreover, a linear model predicting LLM JS divergence as a function of analogy model JS divergence only explains 16% of the variance. While this does not prove that Llama 3 70B Instruct is conducting *bona fide* composition, it provides exciting indications that it might—at minimum, Llama 3 70B Instruct is better able to incorporate the interaction between the adjective meaning and noun meaning than our purely word analogy-based model. Investigating how composition, typically conceptualized as abstract rules, can be implemented in LLMs would be an interesting avenue of future research—the *abstraction-via-exemplars* account discussed in Misra and Kim (2023) may provide a promising starting point.

Standards of evidence for composition This paper contributes to a broader discussion about the standards of evidence required for composition (McCurdy et al., 2024; Pavlick, 2025). If behavioral experiments about generalization can provide evidence about composition (and not all researchers believe they can), we must be sure to rule out other methods of generalization such as analogy. We further need to ensure we have a precise enough definition of compositionality to capture our intuition that analogy, by virtue of referring to information not (obviously) included in the meanings of the parts, is not a kind of composition (Szabó, 2012). By making an explicit model of analogical reasoning, we can both show the way in which it requires this additional information and show that analogical reasoning fails to generalize in the expected way, relative to our human data.

Acknowledgments

Icons in Figure 1 courtesy of The Noun Project, from artists Atacan, Goat_Team, Ricons, vecto and Zaenal Abidin.

References

Kirsten C. S. Adam, Edward K. Vogel, and Edward Awh. 2017. Clear evidence for item limits in visual working memory. *Cognitive Psychology*, 97:79–97.

Adam Albright and Bruce Hayes. 2003. Rules vs. analogy in English past tenses: a computational/experimental study. *Cognition*, 90(2):119–161.

Ben Ambridge. 2020. Against stored abstractions: A radical exemplar model of language acquisition. *First Language*, 40(5-6):509–559.

Inbal Arnon and Neal Snider. 2010. More than words: Frequency effects for multi-word phrases. *Journal of Memory and Language*, 62(1):67–82.

Marco Baroni and Roberto Zamparelli. 2010. Nouns are Vectors, Adjectives are Matrices: Representing Adjective-Noun Constructions in Semantic Space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193, Cambridge, MA. Association for Computational Linguistics.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1):1–48.

Gemma Boleda, Marco Baroni, The Nghia Pham, and Louise McNally. 2013. Intensionality was only alleged: On adjective-noun composition in distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 35–46, Potsdam, Germany. Association for Computational Linguistics.

Mollie E. Brooks, Kasper Kristensen, Koen J. van Benthem, Arni Magnusson, Casper Berg, W., Anders Nielsen, Hans J. Skaug, Martin Mächler, and Benjamin M. Bolker. 2017. glmmTMB Balances Speed and Flexibility Among Packages for Zero-inflated Generalized Linear Mixed Modeling. *The R Journal*, 9(2):378.

Joan L. Bybee. 2010. *Language, usage and cognition*. Cambridge University Press, Cambridge ; New York.

Catherine Caldwell-Harris, Jonathan Berant, and Shimon Edelman. 2012. Measuring Mental Entrenchment of Phrases with Perceptual Identification, Familiarity Ratings, and Corpus Frequency Statistics. In *Measuring Mental Entrenchment of Phrases with Perceptual Identification, Familiarity Ratings, and Corpus Frequency Statistics*, pages 165–194. De Gruyter Mouton.

Bert Cappelle, Denis Pascal, and Mikaela Keller. 2018. Facing the facts of fake: A distributional semantics and corpus annotation approach. *Yearbook of the German Cognitive Linguistics Association*, 6(1):9–42.

Lauretta S. P. Cheng, Danielle Burgess, Natasha Vernooij, Cecilia Solís-Barroso, Ashley McDermott, and Savithry Namboodiripad. 2021. The Problematic Concept of Native Speaker in Psycholinguistics: Replacing Vague and Harmful Terminology With Inclusive and Accurate Measures. *Frontiers in Psychology*, 12.

Nelson Cowan. 2001. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24(1):87–114.

Guillermo del Pinal. 2015. Dual Content Semantics, privative adjectives, and dynamic compositionality. *Semantics and Pragmatics*, 8:7:1–53.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and et al. 2024. The Llama 3 Herd of Models. *arXiv preprint*.

Janek Guerrini. 2024. Keeping Fake Simple. *Journal of Semantics*, 41(2):175–210.

Matthias Hartung, Fabian Kaupmann, Soufian Jebbara, and Philipp Cimiano. 2017. Learning Compositionality Functions on Word Embeddings for Modelling Attribute Meaning in Adjective-Noun Phrases. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 54–64, Valencia, Spain. Association for Computational Linguistics.

Douglas R Hofstadter. 2001. Epilogue: Analogy as the Core of Cognition. *The Analogical Mind*, pages 499–538.

Hans Kamp and Barbara H. Partee. 1995. Prototype theory and compositionality. *Cognition*, 57(2):129–191.

Geoffrey Leech, Paul Rayson, and Andrew Wilson. 2014. *Word Frequencies in Written and Spoken English: based on the British National Corpus*. Routledge, London.

Joshua Martin. 2022. *Compositional Routes to (Non)Intersectivity*. Ph.D., Harvard University, United States – Massachusetts.

Kate McCurdy, Paul Soulos, Paul Smolensky, Roland Fernandez, and Jianfeng Gao. 2024. Toward Compositional Behavior in Neural Models: A Survey of Current Views. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9323–9339, Miami, Florida, USA. Association for Computational Linguistics.

George A. Miller. 1995. WordNet: a lexical database for English. *Commun. ACM*, 38(11):39–41.

Kanishka Misra and Najoung Kim. 2023. Abstraction via exemplars? a representational case study on lexical category inference in bert. In *BUCLD 48: Proceedings of the 48th annual Boston University Conference on Language Development*, Boston, USA.

Neha Nayak, Mark Kowarsky, Gabor Angeli, and Christopher D. Manning. 2014. A Dictionary of Nonsubsective Adjectives. Technical Report CSTR 2014-04, Department of Computer Science, Stanford University.

Barbara H. Partee. 2009. Formal semantics, lexical semantics, and compositionality: The puzzle of privative adjectives. *Philologia*, 7(1):11–21.

Barbara H. Partee. 2010. Privative adjectives: Subsective plus coercion. In Thomas Zimmermann, Rainer Bauerle, and Uwe Reyle, editors, *Presuppositions and discourse: Essays offered to Hans Kamp*, pages 273–285. Brill.

Ellie Pavlick. 2025. Not-Your-Mother's-Connectionism: LLMs as Cognitive Models.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

R Core Team. 2023. R: A Language and Environment for Statistical Computing.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Giulia Rambelli, Emmanuele Chersoni, Philippe Blache, and Alessandro Lenci. 2024. Compositionality as an Analogical Process: Introducing ANNE. In *The First Workshop on Analogical Abstraction in Cognition, Perception, and Language (Analogy-ANGLE)*.

Hayley Ross, Kathryn Davidson, and Najoung Kim. 2024. Is artificial intelligence still intelligence? LLMs generalize to novel adjective-noun pairs, but don't mimic the full human distribution. In *Proceedings of the 2nd GenBench Workshop on Generalisation (Benchmarking) in NLP*, pages 131–153, Miami, Florida, USA. Association for Computational Linguistics.

Hayley Ross, Najoung Kim, and Kathryn Davidson. 2025. Fake reefs are sometimes reefs and sometimes not, but are always compositional. *Experiments in Linguistic Meaning*, 3:332–343.

Zoltán Gendler Szabó. 2012. The case for compositionality. In Wolfram Hinzen, Edouard Machery, and Markus Werning, editors, *The Oxford Handbook of Compositionality*. Oxford University Press.

Antoine Tremblay and Harald Baayen. 2010. Holistic Processing of Regular Four-word Sequences: A Behavioural and ERP Study of the Effects of Structure, Frequency, and Probability on Immediate Free Recall. *Perspectives on Formulaic Language: Acquisition and Communication*, page 151.

Eva M. Vecchi, Marco Marelli, Roberto Zamparelli, and Marco Baroni. 2017. Spicy Adjectives and Nominal Donkeys: Capturing Semantic Deviance Using Compositionality in Distributional Spaces. *Cognitive Science*, 41(1):102–136.

Zhibiao Wu and Martha Palmer. 1994. Verb Semantics and Lexical Selection. In *32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, Las Cruces, New Mexico, USA. Association for Computational Linguistics.

A Analogy Availability for Humans

Figure 7 shows the percentage of times participants were able to find an analogy for each bigram, colored by the estimation of analogy difficulty discussed in Appendix D.

B Participant Recruitment Criteria

For our experiment in Section 5, we recruit people on Prolific who self-report English as their first and primary language and are located in the US. We further ask them at the end of the study whether they learned English before the age of 5 and whether they speak American English—if not, they are paid but excluded from the analysis. This implementation of "native speaker" is merely intended as a practical way to expect shared language experiences among our participant sample (Cheng et al., 2021).

C Detailed Analogy Model Results

C.1 Model Configuration

As discussed in Section 4.1, the model has three configurable parameters: whether to do analogy over just nouns or also to include up to one adjective ("Noun only" vs."Noun + Adjective"), how many nearby bigrams to retain (k), and whether to return the memorized distributions from the training set when asked about a bigram in the training set, or to apply the algorithm as if that particular bigram were not known.

We consider only up to 1 adjective since a hyperparameter search over up to 10 adjectives showed that 1-2 adjectives were always optimal; moreover, we only have 12 candidate adjectives to begin with, and manual inspection suggests that at most 1-2 of them ought to be relevant.

We consider 100 nearby nouns since we do not want to artificially constrain our model and prevent it from finding enough bigrams that it actually knows. Having separate steps for adjective/noun retrieval, assembling candidate bigrams, and then checking which bigrams are known is an artificial implementation choice that we make for our algorithm; humans could well be retrieving similar nouns and checking whether the resulting bigram is known in tandem. Thus, we always retrieve 100 nearby nouns "just in case" and instead rely on the number of bigrams k to constrain the model. As discussed in Section 4.1, we set $k \leq 5$ to impose constraints akin to human working memory (Cowan, 2001; Adam et al., 2017). We allow the model to do a grid search over the exact value of $1 \leq k \leq 5$ by evaluating the model on the training set with memorization disabled. The optimal k typically ranges between 3-5 bigrams. In Table 1, we also report the special configuration $k = 1$, where the model only considers the most similar bigram it can come up with. This mimics humans going with the "first bigram they can come up with", assuming that their retrieval process chooses a good candidate as its first choice.

The final configuration choice, which we did not discuss in Section 4.1, is the training data – what should be considered as bigrams that humans have previously encountered. Option 1 is to include all bigrams classed as "high frequency" by Ross et al. (2024), i.e. all bigrams in the top quartile of their dataset. This results in sparse data for some adjectives. Notably, this only includes a single bigram involving the adjective *knockoff* and no bigrams including *unimportant*, meaning the model will be at a disadvantage for bigrams with these adjectives. In the N+A setting, it will have to rely primarily on bigrams involving e.g. *counterfeit*; in the noun only setting, it will often return no distribution. It is unclear whether this sparsity is precisely realistic, because these adjectives and their bigrams are low-frequency, or not. Options 2a and 2b are to train on the top x most frequent bigrams for each adjective, where we can consider (a) $x = 5$ (akin to the $k \leq 5$ setting for nearby bigrams), or (b) $x = 23$, which results in a nearly identical size training set (276 bigrams) to taking the top quartile (279 bigrams). We report all three

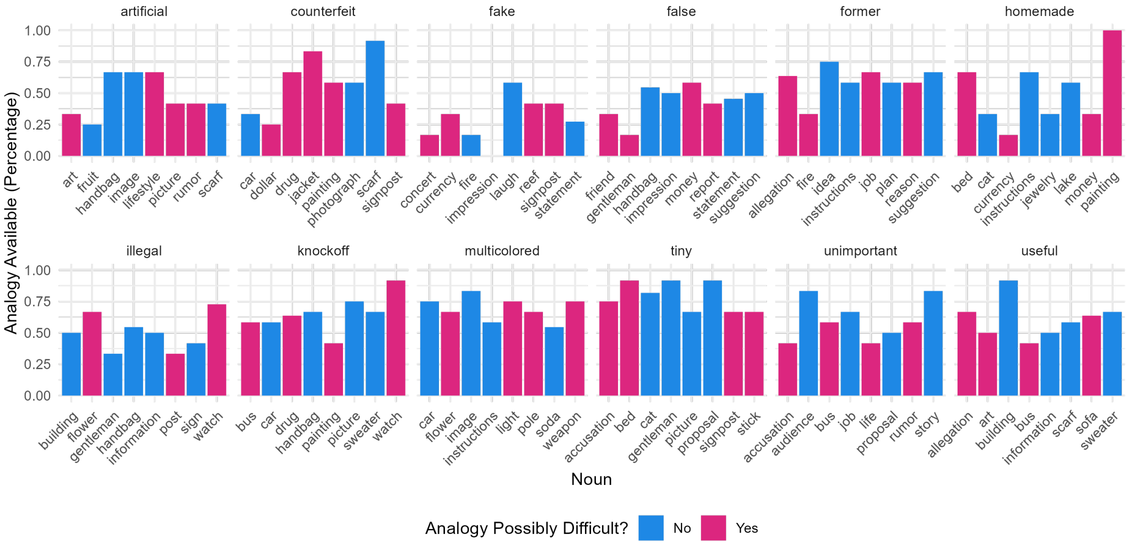


Figure 7: Analogy availability for all 96 bigrams in the analogy prompting experiment. Color indicates whether it was predicted in advance that it might be difficult to find an analogy, based on the ratings from Ross et al. (2025) in conjunction with noun frequencies and WordNet-based distance measures (see Section D).

settings in Table 1.

Finally, in the case where no similar bigrams have known ratings, we opt to return a null distribution, which is always incorrect. We could alternatively return a fallback distribution which concentrates all its probability mass on “Unsure”, but this will also be very unlike the human distributions under the Jensen-Shannon metric (which tend to have high SD when not concentrated at the ends of the scale), so this makes little difference. In practice, this only occurs in the “Noun only” setting for some bigrams involving *knockoff* and *unimportant* when we use the top quartile of bigrams as the training set, since these adjectives have few or no high-frequency bigrams (1 for *knockoff*, 0 for *unimportant*).

C.2 Detailed Results

Table 1 shows the results for the analogy models built with GloVe embeddings, comparing the noun only setting with the N+A (noun + adjective) setting, and the single bigram setting ($k = 1$) with $k \leq 5$. We report the exact value for k chosen by the hyperparameter search. We also compare training on the top quartile of bigrams vs. training on the top 5 or 23 per adjective. Note that for the top 5 case, the set of novel bigrams (column 2, “Novel bigr.”) is larger than in the other cases. We find that the simplest setting, analogy to a single noun (N only, $k = 1$) does not outperform a uniform distribution baseline overall. However, if we allow multiple adjectives, analogy to a single bigram ($k = 1$) is sometimes the best (selected even when we tune on $k \leq 5$). We also achieve similarly good results if we use nouns only but allow averaging over $k \leq 5$ bigrams. In the noun + adjective case, results are also similar whether we train on the top quartile of bigrams or the top 23 bigrams per adjective – training set size appears to be the driving factor, not how it is balanced. However, in the noun only case, which includes all the WordNet models, we unsurprisingly see a performance boost from including more bigrams for each adjective. (When training on the top quartile, the noun only setting necessarily fails for all bigrams involving *unimportant*, since there is no bigram with *unimportant* in the training data, and does poorly for *knockoff* as well, since there is only one bigram with *knockoff* in the training set.) Memorization of the training set boosts overall performance, as expected, though not so much when the training set is very small (top 5 bigrams per adjective).

Further, we observe that performance is generally lower on privative adjectives than overall, which makes sense because many bigrams with subsective adjectives have distributions almost entirely consolidated around “Definitely yes”, and can be predicted from other bigrams.

	JS Divergence (lower is better)				
Model	Novel bigr.	Zero-freq. bigr.	Privative A	Total	Total (+`mem`)
Human (resampled)	N/A	0.04	0.05	0.04	N/A
Human (analogy exp.)	N/A	0.14	0.21	0.16	N/A
Llama 3 70B Instruct	N/A	0.17	0.26	0.17	N/A
Uniform distr. baseline	N/A	0.33	0.20	0.34	N/A
Analogy models: GloVe					
N only, $k = 1$, top qt.	0.44	0.57	0.45	0.39	0.29
N only, $k = 1$, top 5/A	0.32	0.34	0.44	0.32	0.30
N only, $k = 5$, top qt.	0.41	0.55	0.39	0.36	0.27
N only, $k = 3$, top 5/A	0.28	0.28	0.36	0.28	0.25
N only, $k = 4$, top 23/A	0.26	0.25	0.33	0.26	0.17
N+A, $k = 1$, top qt.	0.29	0.31	0.39	0.29	0.19
N+A, $k = 4$, top qt.	0.26	0.26	0.34	0.26	0.17
N+A, $k = 3$, top 5/A	0.27	0.27	0.36	0.27	0.25
N+A, $k = 3$, top 23/A	**0.25**	0.25	**0.32**	0.26	0.17
Analogy models: WordNet					
N only, $k = 1$*, top qt.	0.41	0.54	0.36	0.36	0.26
N only, $k = 1$*, top 23/A	**0.25**	**0.24**	**0.32**	**0.25**	**0.16**
Analogy models: Llama 3 70B embeddings (final layer)					
N only, $k = 1$, top qt.	0.44	0.53	0.44	0.40	0.28
N only, $k = 4$, top qt.	0.40	0.50	0.37	0.35	0.26
N only, $k = 5$, top 23/A	0.26	0.26	0.34	0.26	0.17
N+A, $k = 1$, top qt.	0.33	0.33	0.44	0.34	0.22
N+A, $k = 4$, top qt.	0.28	0.27	0.35	0.28	0.18
N+A, $k = 5$, top 23/A	0.27	0.26	0.34	0.28	0.18
Analogy models: Llama 3 70B embeddings (initial layer)					
N+A, $k = 5$, top qt.	0.28	0.30	0.35	0.27	0.18

Table 1: Average JS divergence (**best** / second) between various configurations of analogy models and human rating distributions, with & without training data memorization, for 'N only' vs. 'N+A' (1 nearby adjective) and $k = 1$ vs. $k \leq 5$ nearby bigrams (exact value of k tuned on training data). 'Novel bigrams' = bigrams held out from each analogy model – for humans and LLMs, we can only be sure that zero-frequency bigrams are novel. 'Privative A' = bigrams with "privative" adjectives. * = set $k \leq 5$ but tuning chose $k = 1$. Llama 3 results and baseline from Ross et al. (2024).

C.3 Significantly Different Distributions

Figure 8 shows 6 of the 10 bigrams where the analogy model (GloVe, $k \leq 5$, with `mem`) predicts a significantly different distribution according to the Kolmogorov-Smirnoff test (with Holm-Bonferroni adjustment) in Section 4.4.

D Estimate of Analogy Difficulty

D.1 Overview

For our analogical reasoning experiment, we attempt to estimate which bigrams might be difficult to find analogies for and balance evenly for this. We suppose that analogy could be difficult for bigrams with one or more following qualities:

- the noun has no high-frequency neighbors (below median among the nouns in the dataset)
- there are multiple convergent nearby bigrams with ratings that conflict
- there are non-convergent nearby bigrams (i.e. bigrams for which the conclusion is uncertain)

We use WordNet (Miller, 1995) rather than word embeddings to find neighboring nouns, since WordNet is manually annotated by human experts, and the British National Corpus for noun frequencies (Leech et al., 2014). We manually define adjective similarity, since WordNet only provides a hierarchical taxonomy – and thus, a similarity metric – over nouns, described in Section D.3.

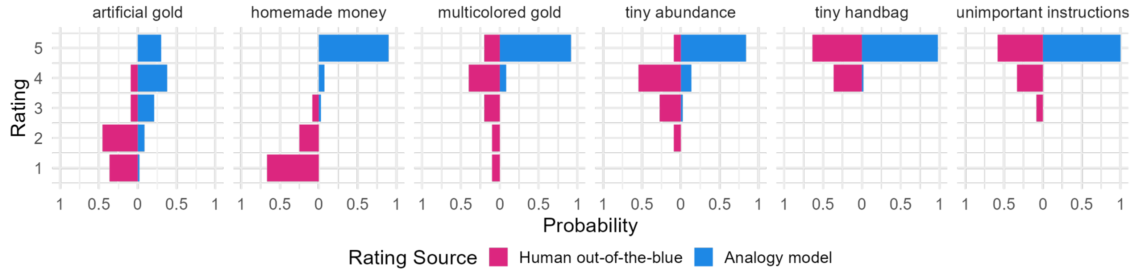


Figure 8: Difference between distributions for 6 of the 10 bigrams which are significantly different between the analogy model (even with `mem`) and the original human distributions. In each case, the model predicts more subsective ratings than humans.

D.2 Results

In fact, we find that these criteria do not predict how often participants were able to come up with an analogy.

We fit a logistic mixed effects model in R (Bates et al., 2015) that predicts whether participants could find an analogy or not. As fixed effects, we include the three factors described in Section 5.1, as well as adjective class (typically privative or subjective) and specificity of the noun (measured by depth in the Wordnet taxonomy). We include adjective and participant as random effects. We also fit a second model where we replace specificity of noun with bigram frequency (the two are too correlated to include in the same model). In fact, we find that none of these factors are significant ($p < 0.05$) except for the presence of nearby divergent bigrams. This feature, however, only applies to 6 bigrams in the experiment, so this may just be spurious. This non-significance may be the result of many false negatives in our labeling of these factors, since we can only test for nearby bigrams among the bigrams that Ross et al. (2025) studied, not among the totality of nearby bigrams. It may also result from our participants construing analogy much more broadly than we did, as discussed in Section 5.2.

D.3 Adjective Similarity Details

We use the following (asymmetric) similarities, which are approximately scaled to match the Wu-Palmer similarity metric (which is 0.5 for siblings).

1. *artificial* → *fake, false*: 0.75
   → *counterfeit, knockoff*: 0.5
2. *counterfeit* → *knockoff*: 0.9
   → *fake, false*: 0.75
   → *artificial*: 0.5
3. *fake* → *artificial, counterfeit, false, knockoff*: 0.75
4. *false* → *fake*: 0.9
   → *counterfeit, knockoff, artificial*: 0.75
5. *knockoff* → *counterfeit*: 0.9
   → *fake*: 0.75
6. *former* → *artificial, counterfeit, fake, false, knockoff*: 0.5
7. *homemade* → *artificial, fake, false*: 0.8
   → *tiny, multicolored*: 0.75
   → *useful, illegal, unimportant*: 0.5
8. The remaining 5 subsective adjectives, *useful, tiny, illegal, unimportant* and *multicolored* are all assigned a similarity of 0.5 to each other and to *homemade*.

Note that we provide an unusually privative-looking set of similarities for *homemade* since the examples with *homemade* in the experiment are disproportionately chosen to be less subsective and thus challenging for analogy. Moreover, these similarities are adjusted for the fact that these are the only 12 adjectives available – of course they would be scaled differently if there were more options. We do not expect small changes to these similarities to have a noticeable difference on the selected bigrams.

E Using Human Analogy Bigrams in the Analogy Model

One bottleneck of our analogy model appears to be its lack of available bigrams with which to draw an analogy, i.e. which it has ratings for, compared to humans. We can try to ameliorate this by additionally giving it all the analogies found in the human analogy experiment, by assuming that the rating that they provide for the target bigram is the same as the rating they would assign to the analogical bigram. (This should be true if they are using the analogy as intended.) We filter the provided analogy phrases through WordNet to retain only two-word phrases whose first word is an adjective and the second a noun. This adds 340 bigrams involving 91 adjectives and 260 nouns. (The original dataset contained only 12 adjectives and 102 nouns.)

Unfortunately, we do not have full distributions for these bigrams; only 68 of the 340 bigrams so found have more than one rating, and only 11 have more than three. For target bigrams with privative adjectives, whose distributions are often spread out, analogy to these new bigrams will thus yield a high JS divergence simply because the distribution is too sparse. In line with this, the results in Table 2 show that adding these additional bigrams worsens or does not improve the two best-performing GloVe models from Table 1, though it does result in different hyperparameter choices during the grid search ($k \leq 5$).

To compensate for only having single ratings, we can instead evaluate the analogy models with the more lenient "accuracy within 1 SD of the human mean" metric proposed for single ratings by Ross et al. (2024), which lets the model predict a mean rating instead of a full distribution. It is then judged "accurate" (enough) if this rating falls within 1 SD of the mean of the human rating distribution that bigram (rounded to the nearest integer), incorrect otherwise. The problem with this metric, besides being ad-hoc, is that the simple "majority" baseline described in Ross et al. (2024), which simply guesses "Unsure" for all bigrams with privative adjectives and "Definitely yes" for all those with subsective adjectives, achieves an accuracy of 0.89 using this metric. Bigrams with privative adjectives generally have such a high SD that this is a large and easy target to hit. Nonetheless, a random guessing baseline scores only 0.46 on this metric, so the metric is still somewhat informative.

If we add the new bigrams provided by the analogy prompting experiment to the training set and evaluate with this Within 1 SD metric, we do see a significant performance increase compared to using just the original training set, as shown in Table 3. Note that optimizing over this metric yields new values for the parameter k, within the constraint $k \leq 5$. $k = 1$ is uniformly chosen during tuning even when we set $k \leq 5$. In contrast to the JS divergence, where we generally saw lower (better) values for subsective adjectives and higher (worse) values for privative ones, this metric yields the opposite, since the SDs for subsective-adjective bigrams are much smaller: we see lower (worse) accuracies for subsective adjectives.

This suggests that if we had full distributions for these bigrams, adding more training data might indeed significantly improve the model. What amount of training data is appropriate for modeling humans remains an open question.

F Experiment Training Instructions

The instructions provided to participants are shown in Table 4.

	JS Divergence (lower is better)				
Model	Novel bigr.	Zero-freq. B	Privative A	Total	Total (+`mem`)
N+A, $k = 4$, top qt.	0.26	0.26	0.34	**0.26**	**0.17**
N+A, $k = 4$, top qt. + exp.	0.45	0.62	0.41	0.39	0.29
N+A, $k = 3$, top 23/A	**0.25**	**0.25**	**0.32**	**0.26**	**0.17**
N+A, $k = 4$, top 23/A + exp.	0.26	0.26	0.33	**0.26**	**0.17**

Table 2: Average JS divergence (**best**) between analogy models and human rating distributions for the best GloVe models in Table 1 and their counterparts trained on the additional bigrams from the human analogy experiment. This additional training data does not improve model performance as measured by JS divergence, because we do not have full distributions for many of the additional bigrams.

	Accuracy within 1 SD of human mean				
Model	Novel bigr.	Zero-freq. B	Privative A	Total	Total (+`mem`)
"Majority" baseline	N/A	0.91	0.78	0.89	N/A
Random guessing baseline	N/A	0.46	0.61	0.46	N/A
N+A, top qt.	0.71	0.77	**0.72**	0.69	0.78
N+A, top qt. + exp.	**0.76**	0.76	0.69	**0.74**	**0.81**
N+A, top 23/A	0.70	0.76	0.71	0.68	0.76
N+A, top 23/A + exp.	0.75	**0.79**	**0.72**	**0.74**	0.80

Table 3: Results for the best GloVe models in Table 1 and their counterparts trained on the additional bigrams from the human analogy experiment using the more lenient "accuracy within 1 SD of human mean" metric proposed by (Ross et al., 2024). All models use $k = 1$ even when tuned with $k \leq 5$; this makes sense as averaging is less likely to improve this metric. Unlike for the JS divergence shown in Table 2, results do improve. However, results must be interpreted relative to the "majority" baseline provided by (Ross et al., 2024), which highlight the difficulty with this metric.

This survey involves questions of the form “Is a toy hippo still large?” We’re interested in whether it’s possible to solve these kinds of questions by reasoning using a similar phrase that you already know the answer for (“by analogy”), such as “toy hippo” → “toy elephant” (toy elephants are usually not large). For the purposes of this survey, the similar phrase / analogy can be another similar thing, or a class of things (like animals or gadgets). The important part is that you know the answer for the new phrase without having to think about it.

Let’s start with three examples that demonstrate how the survey works and what we mean by analogy.

Each question consists of two parts. First you will answer whether you can think of a suitable analogy (yes/no), and type in the similar phrase if you answered yes. The phrase should consist of 1-3 words and will typically be of the form "[adjective] [noun]". Then you will attempt to answer the original question (e.g. "Is a toy hippo still large?") using the phrase you chose, or without it if you couldn’t think of one.

Please pay close attention to the following examples, as we will ask you to follow this style of reasoning in the rest of the survey.

Is melted plastic still plastic?

Can you think of an analogy to another similar phrase that would help answer this question?

You can think of an analogy from “melted plastic” → “melted wax” or “melted chocolate.” This is useful because you immediately know the answer to “Is melted wax still wax?” or “Is melted chocolate still chocolate?” So, you would answer “yes” to this question and type “melted wax” or “melted chocolate” in the text box below.

Based on the analogy you chose:
Is melted plastic still plastic?

Because melted wax is still wax (or melted chocolate is still chocolate), you conclude that melted plastic is still plastic, or probably still plastic. So, you would answer “Definitely yes” or “Probably yes” depending on your interpretation.

Is a hard-boiled egg still runny?

Can you think of an analogy to another phrase that would help answer this question?

You probably find it hard to quickly think of an analogy that can help answer the question. While you may be able to come up with similar phrases, they don’t immediately provide an obvious answer. So, you would answer “No” to this question.
[Instructions for second part irrelevant, omitted]

Is a decorative pumpkin still edible?

Can you think of an analogy to another similar phrase that would help answer this question?

As in the previous example, it is hard to quickly think of an analogy that can help answer the question. While you may be able to come up with similar phrases, they don’t immediately provide an obvious answer. So, you would answer “No” to this question.
[Instructions for second part irrelevant, omitted]

Table 4: Training instructions and examples shown to participants to demonstrate what we intend by “analogy”.

Formalizing Feature Inheritance

Gregory M. Kobele[1] and Lei Liu[2]
Institut für Linguistik
Universität Leipzig
[1]gkobele@uni-leipzig.de,[2]lei.liu@uni-leipzig.de

Abstract

Feature Inheritance is a prominent theoretical innovation in minimalist syntax, which takes it further from the formal framework of minimalist grammars, the best understood formalism for reasoning about minimalism. Feature inheritance involves movement targeting non-root positions, as well as simultaneous movement steps. This turns out to require a formally innocuous extension to minimalist grammars, leaving strong generative capacity and worst-case parsing complexity unchanged.

1 Introduction

Viewing context-free base rules as structure building operations (a rule $S \rightarrow NP\ VP$ builds an S out of a NP and a VP), the transformational cycle in syntax was a principle that governed the interleaving of transformational operations with context-free structure building operations. In particular, (cyclic) transformational rules were applied only once certain categories (always S, often NP, sometimes PP) of expressions were built. In early minimalism, the transformational rule of movement was interleaved with the structure building operation of merge. However, movement could in principle apply at any time, regardless of the categorial status of its input. A mechanism of *feature inheritance*, introduced by Chomsky (2008), in effect delays transformations until a particular category is reached. Thus, minimalism with feature inheritance seems to be a return to the original conception of the syntactic cycle.

In this paper we provide a formalization of the mechanism of feature inheritance in the context of minimalist grammars (MGs), itself a formalization of Chomsky's (1995) Minimalist program. The weak generative capacity and worst-case parsing complexity of *feature inheritance* is then compared to that of vanilla MGs.

2 Feature Inheritance

Minimalist orthodoxy assumes a universal hierarchy of functional projections: Complementizers select Tense which selects Voice which selects Verbs. Underlying these lay terms are the abstract heads (categories) 'C', 'T', 'v' ("little-v"), and 'V' ("big-V"). A large body of work assumes a shared property between little-v and C; these two heads are said to define locality domains in the syntax (called *phases*). A basic goal expressed by Chomsky (1995) is to reduce the stipulations needed in the theory. As little-v and C share one non-trivial property already, determining whether more of their properties can be identified would potentially reduce the number of independent stipulations needed to describe the lexicon. Feature Inheritance (FI) is introduced in (Chomsky, 2008) as a way of reconciling a number of related observations with theoretical assumptions, and is made use of by little-v and C, which increases their formal similarity a great deal.

A main theoretical motivation for FI is to give a larger role to phases. Phases are said to coincide with the portion of the syntactic structure that the interfaces can refer to. In other words, they are the units that semantic and phonological interpretation are defined over. Chomsky suggests that both interfaces refer to the same units of syntactic structure. In addition, he suggests that syntactic operations (like movement and agreement) are not distributed throughout the nodes making up a phase, but are rather deferred until the last head in the phase (little-v or C). This desideratum is problematic from the perspective of orthodox analytical assumptions, as the T head is generally considered to trigger movement of and agreement with the surface subject.

One relevant observation is that only finite T heads trigger movement and agreement. A second observation is that the distribution of finite vs nonfinite T is related to the choice of C: for example, the

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 101-107.
Eugene, Oregon, July 18-20, 2025

declarative complementizer `that` selects for finite T, whereas `for` selects for non-finite T.

1. John believes that Mary smiled.
2. $*$John believes that Mary to smile.
3. $*$John hopes for Mary smiled.
4. John hopes for Mary to smile.

Chomsky's resolution to the problem is to shift the finite-nonfinite distinction over to C, making T into an underspecified tense head. Then it is C which selects for a generic T head, and it must be C which is responsible for triggering movement and agreement **on T**. FI is the mechanism by which movement triggered by a higher head targets the projection of a lower head, which allows for the idea that movement and agreement is deferred until phase heads are introduced to be realized.

C (and little-v) also permit generic movement to their edges, for example, to break long distance movement into phase-sized chunks. Thus C can trigger movement multiple times, both to its edge, as well as to the edge of the T head immediately below it. However, the movements that C now triggers are typically thought to be of two fundamentally different kinds: the movement to T is A-movement, and that targeting C is A-bar-movement. These kinds of movements have importantly different properties (pronouns can be bound after moving over them with A-movement, but not with A-bar-movement, for example), and Chomsky (1995) has proposed that movement steps between the highest A-bar position and the lowest base-merge position of expressions be invisible to various well-formedness conditions. Making the A and A-bar movements which C triggers happen simultaneously (as opposed to serially) structures the movement dependencies entered into by DPs as trees (ordered by derivational order), rather than sequences. This then eliminates the need to postulate an independent operation which deletes intermediate elements in a sequence of movement dependencies — these are no longer on a single branch of the tree.

Feature Inheritance thus paves the way for 1. phase heads to be the locus of movement and agreement triggers, and 2. a novel approach to the distinction between A and A-bar movements.

3 Formal background

We couch our formalization of feature inheritance in the formal framework of minimalist grammars (Stabler, 1997, 2011), an extensible and well-understood grammar formalism capable of transparently representing minimalist analyses. Minimalist grammars are a lexicalized grammar formalism, like categorial grammars, with universal grammatical rules and complex lexical entries. The categories of lexical entries take the form of lists of features, written with lower case greek letters, called *feature bundles*, where a list is a data structure where only the first element is directly accessible. Removing ('checking') the first element of a nonempty list α results in the remainder of the list α' (so $\alpha = a.\alpha'$). Features have one of two polarities (positive and negative), and come in different kinds, represented as different names (k, wh, q, d, $\ldots$). Two features `+x` and `-y` of opposite polarity *match* iff they are of the same kind (i.e. $x = y$).

A syntactic expression is either a pair $\langle w, \alpha\rangle$ consisting of a string of phonemes w and a feature bundle α (written $\mathtt{w}{:}\alpha$), or a term $\bullet(t_1, t_2)$, where t_1 and t_2 are syntactic expressions, and $\bullet$ is either < or >. The *head* of a syntactic expression t is t itself, if a pair, and the head of t_H if $t = \bullet(t_1, t_2)$, where $t_H = t_1$ if $\bullet =$ <, and $t_H = t_2$ if $\bullet =$ >.

Given a syntactic expression t, the result of checking the first feature of its head is written t'. When t is a term, it represents a tree, and the internal nodes 'point' in the direction of the head. A trace is a pair of the empty string and the empty feature bundle, written `t`.

There are two syntactic operations, **Merge** and **Move**. **Merge** is binary, and **Move** unary. They are both restricted in their application by the feature bundles present in their arguments. The head of the first argument of both operations must be a positive feature. **Merge** applies to two expressions t and s just in case the heads of both have matching first features. **Move** applies to its single argument just in case this argument contains a unique leaf whose first feature matches the first feature of the head.

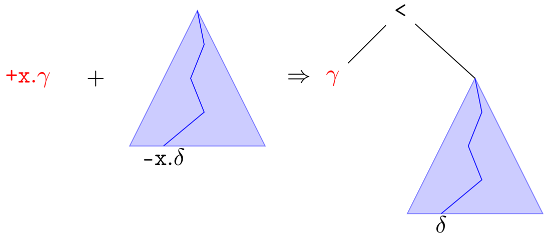


Figure 1: Merge of a complement

The output of **Merge** depends on whether its first argument is a leaf or a complex term. If a leaf ℓ, then **Merge**$(\ell, s) = <(\ell', s')$, and if a proper term t, **Merge**$(t, s) = >(t', s')$, as is depicted in figures 1 and 2.

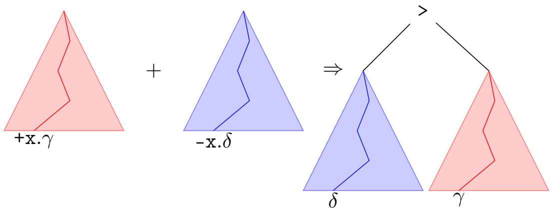


Figure 2: Merge of a specifier

Move replaces a subterm of the input with a trace, and so we need a notation which simplifies referring to subterms. We define *maximal projection contexts* $C[x]$ to be either a variable x, or a structure of one of the two forms: $>(C[x], t)$ or $<(t, C[x])$. A maximal projection context $C[x]$ is a term where x occurs without any arrows pointing to it, and replacing the variable x with a term s is written $C[s]$. **Move** applies to t iff $t = C[s]$, where s is a term whose head begins with a negative feature which matches that of t. **Move**$(C[s]) = >(s', C[\mathsf{t}]')$, as is depicted in figure 3.

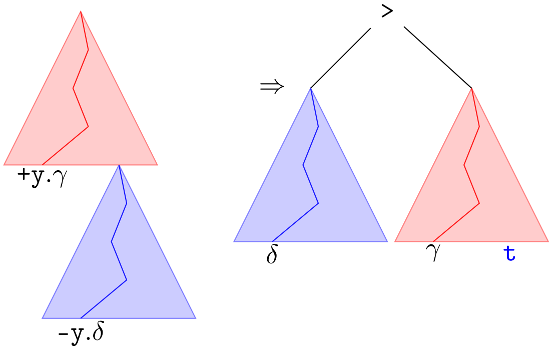


Figure 3: Movement leaves a trace

Both operations have the effect of removing features from feature bundles one at a time, and features in feature bundles are checked one at a time from left to right.

4 Features for Feature Inheritance

Feature inheritance diverges from minimalist grammars as they have been defined above in two ways. First, movement can target not the top of an expression, but rather some node embedded inside it. Second, two features can be checked at the same time.

To deal with the first difference, we allow positive features to take a diacritic (written: $\mathsf{+x}^{\downarrow}$) indicating that they should target the sister node to the head. We can augment the **Move** operation so that it can deal with these new feature types. For example, given a term t the first feature of the head of which begins with $\mathsf{+y}^{\downarrow}$, whose complement $C[s]$ contains a unique term s with matching first feature, write $t = D[C[s]]$. Then **Move**$(D[C[s]]) = D[>(s', C[\mathsf{t}])]'$. This is shown in figure 4.

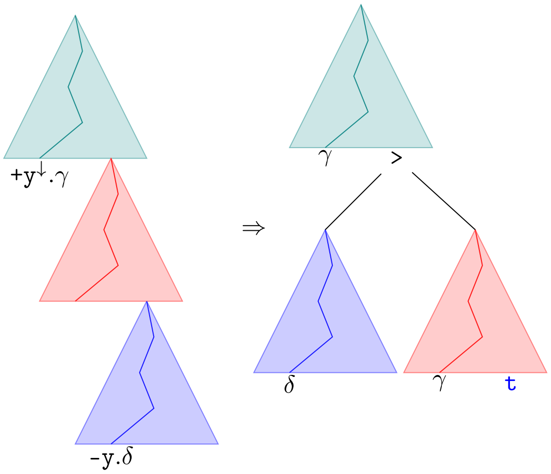


Figure 4: Inherited movement

To allow two features to be checked simultaneously, we allow feature bundles to contain not just individual features, but also pairs of features. Given a pair of features $\langle\mathsf{+x}, \mathsf{+y}\rangle$, it is intended that they be checked during the same derivational step. This allows us to write lexical items with the desired behaviour; Chomsky's C head would have feature bundle $\mathsf{+T}.\langle\mathsf{+k}^{\downarrow}, \mathsf{+wh}\rangle.\mathsf{-C}$, indicating that it first merges with a TP, after which it simultaneously triggers k-movement to TP and wh-movement to itself, and then is itself a CP. Introducing two new feature types ($\mathsf{+x}^{\downarrow}$ and $\langle f, g\rangle$) would allow for lexical feature bundles of the following forms:

1. $\mathsf{+a}.\mathsf{+b}.\mathsf{+c}^{\downarrow}.\mathsf{-d}$

2. $\mathsf{+a}.\mathsf{+b}^{\downarrow}.\mathsf{+c}^{\downarrow}.\mathsf{-d}$

3. $\mathsf{+a}.\langle\mathsf{+b}, \mathsf{+c}\rangle.\mathsf{-d}$

These bundles express sequences of lexically driven derivational steps which we view as not in the spirit of Chomsky (2008), which we summarize with the following principles:

FIUniq Feature inheritance happens just once

FIEarly Feature inheritance happens immediately after the complement is merged

FISimul Simultaneous feature checking happens only in the context of feature inheritance

Feature bundle 1 violates the earliness principle (**FIEarly**), which requires feature inheritance to happen immediately after the complement is merged. Here, feature inheritance of $\texttt{+c}^{\downarrow}$ was deferred until after $\texttt{+b}$ was checked. Feature bundle 2 violates both the uniqueness principle (**FIUniq**), which requires feature inheritance to occur just once, and the earliness principle. Here, feature inheritance occurs both via $\texttt{+b}^{\downarrow}$ and $\texttt{+c}^{\downarrow}$, and in addition $\texttt{+c}^{\downarrow}$ was deferred until after $\texttt{+b}^{\downarrow}$ was checked. Feature bundle 3 violates the simultaneity principle (**FISimul**), which requires that simultaneous feature checking occur in conjunction with feature inheritance. Here, features $\texttt{+b}$ and $\texttt{+c}$ are checked simultaneously, neither of which involve feature inheritance. These principles conspire to enforce that lexical feature bundles are drawn from the following regular set, where $\mathbb{P} := \{\texttt{+x} \mid x \in \mathbb{F}\}$, $\mathbb{D} := \{\texttt{+x}^{\downarrow} \mid x \in \mathbb{F}\}$, $\mathbb{S} := \{\langle d, p\rangle \mid d \in \mathbb{D} \wedge p \in \mathbb{P}\}$ and $\mathbb{N} := \{\texttt{-x} \mid x \in \mathbb{F}\}$:

$$(\mathbb{P}(\mathbb{D}+\mathbb{S})^{?})^{?}\mathbb{P}^{*}\mathbb{N}^{+}$$

That is, an inheritance feature occurs only after the first positive feature, either on its own or as part of a simultaneous feature. With respect to the requirement that exactly one of the pair of simultaneous features must be an inheritance feature has a certain coherence to it. Note that with any other combination of simultaneous features (i.e. where both are of the same kind) it would be unclear how to depict the derived tree which should result after the simultaneous features are checked: as both target the same position (either the complement to the head, or the specifier of the same) one mover would need to c-command the other, from which one could reconstruct a checking order, belying the simultaneity of checking.

5 Implementing Feature Inheritance

A naïve implementation of inherited movement as in figure 4 is destructive, in the sense that constructing the output requires changing immediate dominance relations which held in the input. (In particular, the immediate dominance between the mother '<' of the head of the tree and the root of its complement.) For reasons discussed in the next section, this is to be avoided when possible.

Taken together, the constraints on feature bundles presented above allow for an alternative implementation of feature inheritance. As feature inheritance targets the first merged argument of the head, and takes place immediately after this argument is merged, it is simple to deal with feature inheritance during this very **Merge** step, where the top of the second argument is still accessible. This avoids the problem of destructivity, as the target position of the inherited movement has not yet been assigned an immediate dominance relation. Let ℓ be a lexical item whose feature bundle begins with the following two features: $\texttt{+x}$ and $\langle \texttt{+y}^{\downarrow}, \texttt{+z}\rangle$. There are two cases to consider, depending on whether one mover matches both features in the pair, or whether they are matched by different movers. For the first case, let $C[s]$ be a term with first feature $\texttt{-x}$, and where the first two features of s are $\texttt{-y}$ and $\texttt{-z}$. Then $\textbf{Merge}(\ell, C[s]) = \texttt{>}(s'', \texttt{<}(\ell'', \texttt{>}(\texttt{t}, C[\texttt{t}]')))$, as is depicted in figure 5. In the other case, let $C[x, y]$ be a maximal projection context with *two* variables, and let $C[r, s]$ be a term whose first feature is $\texttt{-x}$, and where the first features of r and s are $\texttt{-y}$ and $\texttt{-z}$ respectively. Then $\textbf{Merge}(\ell, C[r, s]) = \texttt{>}(s', \texttt{<}(\ell'', \texttt{>}(r', C[\texttt{t}, \texttt{t}]')))$, as is depicted in figure 6.

It only really matters that the movement steps be simultaneous if the same mover is targeted in both cases. This is because Chomsky analyzes the twin movements as creating different chains — sequential movements of the same item would simply extend a single chain. If two different movers are targeted, each is going to extend its own chain, regardless of whether this happens simultaneously or sequencially.

6 Complexity Analysis

Michaelis (2001) (see also Harkema (2001)) proves the equivalence between minimalist grammars and multiple context-free grammars, providing a scaffolding for future demonstrations that extensions do not increase generative capacity. To establish such an equivalence, we need to present the modified operations in inference rule format, stated over finite sequences of strings paired with feature bundles. As noted by Stanojević (2019), parsers

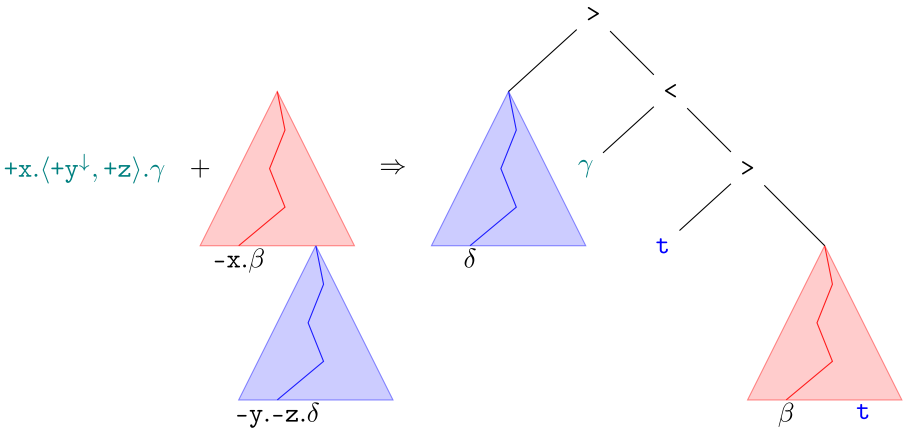


Figure 5: Feature inheritance involving a single mover

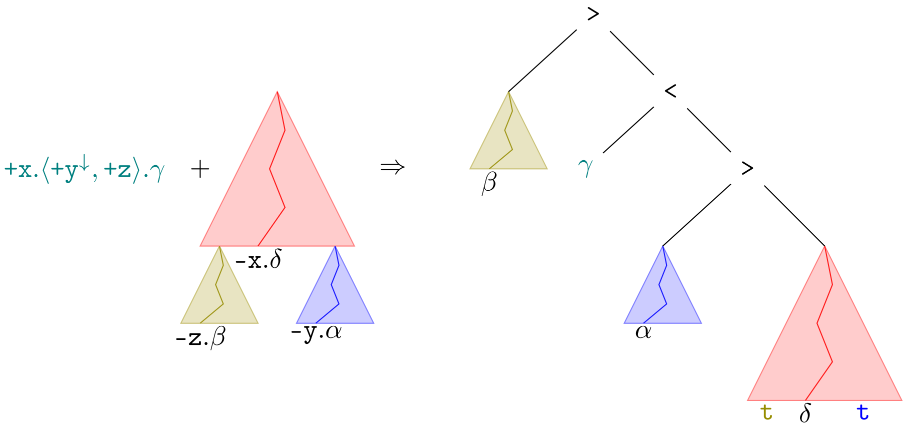


Figure 6: Feature inheritance involving two different movers

derived from this inference rule notation can have their worst-case time complexity read directly off of the rules themselves. Representing each string as a *span*, a pair of integer variables indicating what portion of the input string that string should cover, the number of distinct variables in the antecedents of a rule polynomially bounds its contribution to worst case complexity. Our revised implemention of feature inheritance only modifies the **Merge** rule (by adding to it two new cases), and so we present just these in inference rule format (see Stabler and Keenan (2003) for the others). In inference rule notation, to each term corresponds a sequence of string-feature bundle pairs. Each pair beyond the first corresponds to a maximal proper subterm whose head begins with negative features. The first pair corresponds to the term minus these moving pieces.

The inference rules are given in the figures 7–12. This summation and the associated computational complexity is indicated next to the names of each of the rules above. We see that the rules **MrgFI1b** and **MrgFI2d** contribute the most to the worst case time complexity of the new rules. To put this in perspective, the worst case time complexity of minimalist grammars *without* feature inheritance is also $\mathcal{O}(n^{2k+3})$ (Fowlie and Koller, 2017; Stanojević, 2019). Thus minimalist grammars with feature inheritance have the same worst case time complexity as vanilla MGs.

7 Conclusion

We have presented a formalization of Chomsky's ((2008)) mechanism of feature inheritance, which has played an important role in minimalist syntactic theory over the intervening nearly two decades. It is formally innocuous: it increases neither the weak generative capacity nor the worst case time complexity of the MG formalism.

Another route to this result is to simply note that lexica containing the new lexical items with feature bundles of the form $\mathsf{+x.+y}^{\downarrow}.\alpha$ and $\mathsf{+x}.\langle\mathsf{+y}^{\downarrow},\mathsf{+z}\rangle.\alpha$ can be transformed into strongly and weakly equivalent lexica containing only standard feature bundles: given a lexical item u:$\mathsf{+x}.\langle\mathsf{+y}^{\downarrow},\mathsf{+z}\rangle.\alpha$, replace it with a lexical item u:$\mathsf{+x'.+z}.\alpha$, where x' is a fresh feature name, and for every lexical item v:$\beta.\mathsf{-x}.\gamma$ add to the lexicon the new lexical item v:$\beta.\mathsf{+x.-x'}.\gamma$. This transformation simply pushes down the inherited features onto the lexical items which will ultimately inherit them, and ensures that they subsequently combine with their benefactors.

Like many proposals in minimalism, the substance of this one seems to lie in things not so easily measured, like: 1. providing a formal foundation for the distinction between movement types: two independent chains branching off of a single element, one of which c-commands the other, gives a scaffolding over which different clusters of properties can be assigned to each, and 2. giving a formal unification of lexical items of a certain type: $\forall\mathsf{x.+x}.\langle\mathsf{+}\phi^{\downarrow},\mathsf{+epp}\rangle.\mathsf{-x'}$ is the general format for phasal heads, where *epp* is a feature permitting movement, and ϕ are agreement related features (and we have used object-level quantification over feature names to express polymorphism, and x' is the next category up in the extended projection of x).

References

Noam Chomsky. 1995. *The Minimalist Program*. MIT Press, Cambridge, Massachusetts.

Noam Chomsky. 2008. On phases. In Robert Freidin, Carlos P. Otero, and Maria Luisa Zubizarreta, editors, *Foundational Issues in Linguistic Theory*, pages 133–166. MIT Press, Cambridge, Massachusetts.

Meaghan Fowlie and Alexander Koller. 2017. Parsing minimalist languages with interpreted regular tree grammars. In *Proceedings of the 13th International Workshop on Tree Adjoining Grammars and Related Formalisms*, pages 11–20, Umeå, Sweden. Association for Computational Linguistics.

Henk Harkema. 2001. *Parsing Minimalist Languages*. Ph.D. thesis, University of California, Los Angeles.

Jens Michaelis. 2001. *On Formal Properties of Minimalist Grammars*. Ph.D. thesis, Universität Potsdam.

Edward P. Stabler. 1997. Derivational minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–95. Springer-Verlag, Berlin.

Edward P. Stabler. 2011. Computational perspectives on minimalism. In Cedric Boeckx, editor, *The Oxford Handbook of Linguistic Minimalism*, Oxford Handbooks in Linguistics, chapter 27, pages 616–641. Oxford University Press, New York.

Edward P. Stabler and Edward L. Keenan. 2003. Structural similarity within and among languages. *Theoretical Computer Science*, 293:345–363.

Miloš Stanojević. 2019. On the computational complexity of head movement and affix hopping. In *Formal Grammar*, pages 101–116, Berlin, Heidelberg. Springer Berlin Heidelberg.

$$\frac{\langle m, \mathsf{+x.(+y^{\downarrow}, +z)}.\alpha\rangle \qquad \langle n, \mathsf{-x}\rangle, \vec{\phi}, \langle o, \mathsf{-y.-z}\rangle, \vec{\psi}}{\langle omn, \alpha\rangle, \vec{\phi}, \vec{\psi}}\ \text{MrgFI1a} \qquad \mathcal{O}(n^{2k+2})$$

The inference rule **MrgFI1a** describes the situation where there is a single mover, for whom this is the last movement step, and therefore is pronounced in its highest position.

Figure 7: **MrgFI1a**

$$\frac{\langle m, \mathsf{+x.(+y^{\downarrow}, +z)}.\alpha\rangle \qquad \langle n, \mathsf{-x}\rangle, \vec{\phi}, \langle o, \mathsf{-y.-z}.\beta\rangle, \vec{\psi}}{\langle mn, \alpha\rangle, \vec{\phi}, \langle o, \beta\rangle, \vec{\psi}}\ \text{MrgFI1b} \qquad \mathcal{O}(n^{2k+3})$$

The inference rule **MrgFI1b** describes the situation where the single mover has features left over, and thus continues moving.

Figure 8: **MrgFI1b**

$$\frac{\langle m, \mathsf{+x.(+y^{\downarrow}, +z)}.\alpha\rangle \qquad \langle n, \mathsf{-x}\rangle, \vec{\phi}, \langle o, \mathsf{-y}\rangle, \vec{\psi}, \langle p, \mathsf{-z}\rangle, \vec{\chi}}{\langle pmon, \alpha\rangle, \vec{\phi}, \vec{\psi}, \vec{\chi}}\ \text{MrgFI2a} \qquad \mathcal{O}(n^{2k+1})$$

The inference rule **MrgFI2a** describes the situation where there are two movers, for both of which this is the last movement step, and therefore are pronounced in their highest positions. In the result, we see that the phonetic part *o* of the tucking-in mover is sandwiched between the head *m* selecting the complement, and the pronunciation *n* of this complement.

Figure 9: **MrgFI2a**

$$\frac{\langle m, \mathsf{+x.(+y^{\downarrow}, +z)}.\alpha\rangle \qquad \langle n, \mathsf{-x}\rangle, \vec{\phi}, \langle o, \mathsf{-y}.\beta\rangle, \vec{\psi}, \langle p, \mathsf{-z}\rangle, \vec{\chi}}{\langle pmn, \alpha\rangle, \vec{\phi}, \langle o, \beta\rangle, \vec{\psi}, \vec{\chi}}\ \text{MrgFI2b} \quad \mathcal{O}(n^{2k+2})$$

The inference rule **MrgFI2b** describes the situation where there are two movers, but the first one continues moving.

Figure 10: **MrgFI2b**

$$\frac{\langle m, \mathsf{+x.(+y^{\downarrow}, +z)}.\alpha\rangle \qquad \langle n, \mathsf{-x}\rangle, \vec{\phi}, \langle o, \mathsf{-y}\rangle, \vec{\psi}, \langle p, \mathsf{-z}.\gamma\rangle, \vec{\chi}}{\langle mon, \alpha\rangle, \vec{\phi}, \vec{\psi}, \langle p, \gamma\rangle, \vec{\chi}}\ \text{MrgFI2c} \quad \mathcal{O}(n^{2k+2})$$

The inference rule **MrgFI2c** describes the situation where there are two movers, but the second one continues moving.

Figure 11: **MrgFI2c**

$$\frac{\langle m, \mathsf{+x.(+y^{\downarrow}, +z)}.\alpha\rangle \qquad \langle n, \mathsf{-x}\rangle, \vec{\phi}, \langle o, \mathsf{-y}.\beta\rangle, \vec{\psi}, \langle p, \mathsf{-z}.\gamma\rangle, \vec{\chi}}{\langle mn, \alpha\rangle, \vec{\phi}, \langle o, \beta\rangle, \vec{\psi}, \langle p, \gamma\rangle, \vec{\chi}}\ \text{MrgFI2d} \quad \mathcal{O}(n^{2k+3})$$

The inference rule **MrgFI2d** describes the situation where there are two movers, and both continue moving.

Figure 12: **MrgFI2d**

BMRS-Net: Learning BMRS Predicates as Decision Trees

Yifan Hu[*]
University College London
yifanhu@umass.edu

Abstract

This paper explores two applications of learning Boolean Monadic Recursive Scheme (BMRS) feature predicates, leveraging their analogy to binary Decision Trees. Through two case studies, the paper demonstrates how these applications can successfully fit some datasets and analyze new phonological transformations in a decision-based approach, while retaining high transparency and interpretability.

1 Introduction

Phonology has traditionally been guided by frameworks such as the Sound Pattern of English (SPE) and Optimality Theory (OT) to understanding transformations and constraint satisfaction (Chomsky and Halle, 1968; Prince and Smolensky, 2002). However, there is an increasing interest in more computationally oriented models that can handle large datasets and adapt dynamically to new linguistic context. One such model is the Boolean Monadic Recursive Scheme (BMRS), a decision-based approach that utilizes recursive functions and Boolean logic, making it particularly compatible for extensive phonological analysis (Bhaskar et al., 2020; Chandlee and Jardine, 2021).

BMRS is structured around "if-then-else" expressions, which resemble the nodes of a binary Decision Tree where each decision leads to further branches and conditions. This decision-based structure associates it closely with computational models used in data science and machine learning (Quinlan, 1986). While BMRS predicates were typically defined manually (e.g., Hua et al., 2021; Oakden, 2021; Zhu, 2023; Jardine and Oakden, 2023), recent work demonstrates that decision tree learning algorithms can classify and stratify contrastive phonological features accordingly (Chandlee, 2023), suggesting a potential for these algorithms to automate the learning of BMRS feature predicates.

This paper employs the Classification and Regression Trees (CART) algorithm as a tool for automating the generation of BMRS feature predicates (Breiman et al., 1984; Pedregosa et al., 2011; Geron, 2019). We conceptualize a type of binary decision trees, termed *BMRS-Trees*, where the root and each intermediate node utilize only one Boolean attribute. Additionally, by connecting multiple BMRS-Trees in parallel, we can output a comprehensive phonological feature matrix – this network-like structure is termed the *BMRS-Net*.

2 Preliminaries

2.1 Boolean Monadic Recursive Schemes (BMRS)

BMRS (Bhaskar et al., 2020; Chandlee and Jardine, 2021) can best be conceptualized as an **index-by-index UR-to-SF** (Underlying Representation-to-Surface Form) **transducer**. It processes each index individually, starting from index 1 and iterating rightwards. To illustrate, in then mapping from the input string $x_1x_2 \ldots x_N$ to the output string $y_1y_2 \ldots y_N$, index 1 is the first to be assessed by BMRS' feature predicate, returning a Boolean value $True$ (denoted by $\top$ in this paper, or numeric 1 in vectors and matrices) or $False$ ($\bot$ or 0) that determines the output y_1, then index 2, index 3, until N. Each output character y_i is produced based primarily on its corresponding input character x_i, and the whole input string also provides contextual information, as well as all the y_i's predecessors in the output string. Given its index-by-index nature, BMRS requires its input and output be of the same length for error-free index iteration.[1]

[*] Research conducted while the author was affiliated with University College London. The author will begin a PhD program at University of Massachusetts Amherst in September 2025. This paper is a revised version of his MA dissertation.

[1] Readers unfamiliar with BMRS transduction may refer to Appendix A for a running example after finishing 2.1.

BMRS utilizes two position functions to navigate and manipulate string indices: the **predecessor** p and **successor** s, defined for any index i in a string of length N as:

$$p(x_i) = \begin{cases} x_{i-1} & \text{if } i > 1 \\ \_ & \text{if } i = 1 \end{cases}$$

$$s(x_i) = \begin{cases} x_{i+1} & \text{if } i < N \\ \_ & \text{if } i = N \end{cases}$$

The underscore _ serves as a boundary symbol at both ends.[2]

Recursively nesting position functions allows access to any preceding and succeeding characters of any given index, indicated by superscripts, e.g.:

$$p^2(x_i) = p(p(x_i))$$

$$s^3(x_i) = s(s(s(x_i)))$$

A superscripted asterisk ($*$) indicates an arbitrary number of nestings, e.g.: [3]

$$p^*(x_i) = p(x_i), p(p(x_i)) \text{ or } p(p(p(\ldots(x_i))))$$

Σ denotes the **Symbol Set** or **Alphabet**, encompassing all characters all possible characters in both input and output strings; the modified Σ_ incorporates the boundary symbol _. Feature predicates for each symbol σ in Σ are defined as:

$$\sigma(x) = \begin{cases} \top & \text{if } x \vDash \sigma \\ \bot & \text{if } x \nvDash \sigma \end{cases}$$

These feature predicates assess whether the character x at the current index "satisfies" or "models" the symbol σ, returning either $\top$ or $\bot$.[4] When applied to output strings, they are subscripted with an o to differentiate their application context, e.g.:[5]

$$\sigma_o(x_i) = \begin{cases} \top & \text{if } y_i \vDash \sigma \\ \bot & \text{if } y_i \nvDash \sigma \end{cases}$$

A well-formed BMRS predicate might include:

1. **Symbolic feature predicates**, which are simple checks like $\sigma(x)$ that directly assess the "match" of a symbol at the current index;
2. **Position-embedded feature predicates**, more complex predicates like $\sigma(p(x))$, $\sigma(s^2(x))$ or $\sigma(p^*(x))$[6] that evaluate the "match" of symbols at positions relative to the current index; and
3. **Conditional logic**, which refers to construction of "if-then-else" statements upon symbolic feature predicates, position-embedding feature predicates, and $\top/\bot$, e.g., if $\sigma(x)$ then $\top$ else $\sigma(p(x))$.

2.2 Decision Tree

…is a supervised learning model is used for classification and regression tasks (Quinlan, 1986; Breiman et al., 1984). It recursively splits the dataset based on input attributes, forming a tree where each node represents a decision, and each branch corresponds to a possible outcome. The leaf nodes return the predicted output.

This paper focuses on **binary classification decision trees**, where all attributes (including the target attribute) are Boolean. The training data is typically in a table, with each row representing a data instance and each column an attribute for splitting. The last column is by convention the target attribute, which the Decision Tree aims to predict. An example table is presented in Table 1:

Attribute 1	Attribute 2	Attribute 3	Target
$\top$	$\top$	$\bot$	$\bot$
$\top$	$\top$	$\top$	$\bot$
$\top$	$\bot$	$\bot$	$\bot$
$\top$	$\bot$	$\top$	$\bot$
$\bot$	$\top$	$\bot$	$\top$
$\bot$	$\top$	$\top$	$\top$
$\bot$	$\bot$	$\bot$	$\bot$
$\bot$	$\bot$	$\top$	$\top$

Table 1: Example Attribute Table

Our implementation uses the scikit-learn library (Pedregosa et al., 2011), which defaults to the Classification and Regression Tree (CART) algorithm for growing Decision Trees (Breiman et al., 1984). When applied to the dataset in Table 1, CART generates the Decision Tree shown in Figure 1:

[2] The standard implementation uses a left edge ($\rtimes$) and right edge marker ($\ltimes$) instead (Bhaskar et al., 2020; Chandlee and Jardine, 2021), but in this paper use of _ is equivalent.

[3] The asterisk notation is an ad hoc reader-friendly simplification. Precise definition will be give in Footnote 6.

[4] Σ can denote beyond mere "symbols": when Σ denotes a set of phonological features (e.g., [front], as in Section 3.2) and x denotes some segment (e.g., [i]), then saying "[i] satisfies (or $\vDash$) [front]" makes more sense.

[5] $\sigma_o(x_i)$ is a target feature predicate, see Section 3.1.

[6] Technically, $p^*(x)$ is undefined and not a formal term used in BMRS. Below, we will first provide a precise definition of the functions p^* and s^*:

$p^*(\sigma, x_i) = \text{if } \sigma(x_i) \text{ then } \top \text{ else (if } \_(x_i) \text{ then } \bot \text{ else } p^*(\sigma, p(x)) \text{)}$
$s^*(\sigma, x_i) = \text{if } \sigma(x_i) \text{ then } \top \text{ else (if } \_(x_i) \text{ then } \bot \text{ else } s^*(\sigma, s(x)) \text{)}$

For simplicity, we will write both functions as $\sigma(p^*(x))$ and $\sigma(s^*(x))$ in the rest of the paper.

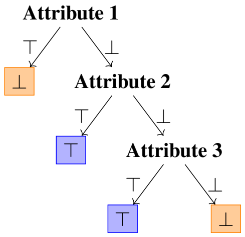


Figure 1: Example Decision Tree

While the Tree-growing algorithm is well-established (see Appendix B for a detailed explanation of CART), our focus will primarily revolve around extracting robust *attributes* (*feature predicates* in the context of BMRS, see Section 3.1).

3 BMRS-Tree

3.1 Implementation

As BMRS calculates output feature predicates index-by-index, extracting features for the CART attribute table also requires index-by-index processing. Given Σ and a UR $x_1x_2\ldots x_N$ to SF $y_1y_2\ldots y_N$ mapping, we propose that **at each index** i the following categories of feature predicates be aggregated:

Symbolic Feature Predicates: These represent whether an input character x_i matches each symbol σ in Σ, denoted as:

$$A_{symbolic} = \{\sigma(x_i) \mid \sigma \in \Sigma\}$$

Local Feature Predicates: To capture phonological dependencies from adjacent symbols, we define local feature predicates within a length of scanning window L(cf. Hua et al., 2021), with $L = 2$ by default:

$$\begin{aligned} A_{local} &= \{\sigma(p^k(x_i)) \mid \sigma \in \Sigma\_, 1 \leq k \leq L\} \\ &\cup \{\sigma(s^k(x_i)) \mid \sigma \in \Sigma\_, 1 \leq k \leq L\} \end{aligned}$$

Here, _ helps BMRS capture the absolute distance from the boundary, such as $\_(p(x_i))$ denoting whether x_1 is the first character, or $\_(s^2(x_i))$ denoting whether x_i is penultimate.[7]

Global Feature Predicates: These capture long-distance dependencies by scanning bidirectionally through the input, without precise index positioning:

$$\begin{aligned} A_{global} &= \{\sigma(p^*(x_i)) \mid \sigma \in \Sigma\} \\ &\cup \{\sigma(s^*(x_i)) \mid \sigma \in \Sigma\} \end{aligned}$$

Output-Dependent Feature Predicates: Based on Output Strictly-Local (OSL) transformations identified by (Chandlee, 2014) (see also Chandlee and Jardine, 2014; Chandlee et al., 2015, 2018), these predicates focus on the most recent output. We define two sets, local and global:

$$\begin{aligned} A_{localOutput} &= \{\sigma_o(p^k(x_i)) \mid \sigma \in \Sigma\_, 1 \leq k \leq L\} \\ A_{globalOutput} &= \{\sigma_o(p^*(x_i)) \mid \sigma \in \Sigma\} \end{aligned}$$

It's worth noting that output-dependent predicates $A_{localOutput}$ and $A_{globalOutput}$ differ from input-related A_{local} and A_{global} by including only one position function p, meaning they are restricted to left-subsequential. Unlike (Oakden, 2021), which used both left- and right-subsequential OSL functions, this paper prohibits right-subsequential OSL functions to avoid backtracking, in line with the no-backtracking mechanism of BMRS. Once an index returns an output, none of its predecessors can be reevaluated to adjust earlier outputs. Similar to A_{local} and A_{global}, $A_{localOutput}$ and $A_{globalOutput}$ embed an accurate memory of a length L scanning window and a vague memory of long-distance dependencies with respect to the output, essential for modeling phonological transformations where previous outputs influence the current index.

Target Feature Predicates: This category contains Boolean representations of the current index's output, with each feature predicate within it serving as the target attribute (the last column of an attribute table) and represented as a BMRS-Tree, denoted as:

$$A_{target} = \{\sigma_o(x_i) \mid \sigma \in \Sigma\}$$

A visual demonstration of the aggregation of feature predicates is presented in Figure 2, where each category is labeled with its number and set name, with superscripts p or s denoting left- or right-subsequential categories (cf. Oakden, 2021); application scopes are indicated by solid lines (accurate memory) or dashed lines (vague memory). All categories except A_{target} (1 to 4) constitute the set A of "Attributes" in Section 2.2:

$$A = A_{symbolic} \cup A_{local} \cup A_{global} \cup A_{localOutput} \cup A_{globalOutput}$$

[7] Strictly speaking, $\_(s^2(x_i))$ does not express "current index i being penultimate" with complete accuracy: supposing i already being final, then its successor of successor is still the boundary symbol _. Hence, in this paper, every position-embedded feature predicate with respect to _ inherently carries a second check that its predecessor/successor is not the boundary symbol _ (see below). But for simplicity, we still use $\_(s^2(x_i))$ to denote $penult(x_i)$ in the rest of the paper.

$$\text{penult}(x_i) = \text{if } \_(s^2(x_i)) \text{ then (if } \_(s(x_i)) \text{ then } \bot \text{ else } \top) \text{ else } \bot$$

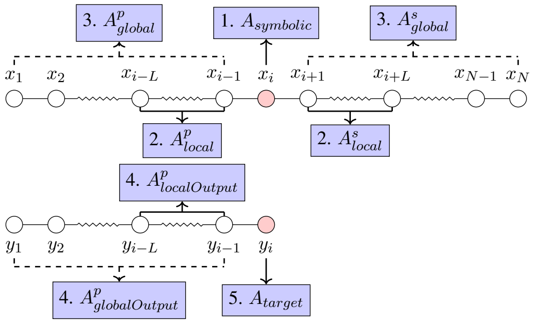


Figure 2: Aggregation of Feature Predicates at index i

The procedure of aggregating feature predicates is to arrange all the instances we extract into an attribute table, as displayed in Table 2, where the header row contains the names of each attribute $a \in A$, with each attribute a_i being an individual column. By convention the last header corresponds to one target attribute $a_{target} \in A_{target}$, the target BMRS-Tree to be learned from this attribute table.

	a_1	a_2	a_3	$\cdots$	$a_{\|A\|}$	a_{target}
Idx 1 of Data 1	$\top$	$\top$	$\bot$	$\cdots$	$\bot$	$\top$
Idx 2 of Data 1	$\top$	$\bot$	$\bot$	$\cdots$	$\top$	$\top$
$\vdots$				$\ddots$		$\vdots$
Idx N of Data 1	$\bot$	$\top$	$\top$	$\cdots$	$\bot$	$\bot$
Idx 1 of Data 2	$\top$	$\bot$	$\bot$	$\cdots$	$\bot$	$\top$
Idx 2 of Data 2	$\bot$	$\bot$	$\bot$	$\cdots$	$\top$	$\bot$
$\vdots$				$\ddots$		$\vdots$

Table 2: Example Attribute Table for BMRS-Tree

For the table content, **every row is filled in with an instance extracted from one certain index within a certain piece of data**, which represents a comprehensive snapshot of the phonological states around that index position in the string. The attribute table grows iteratively as we traverse through all possible indices across every piece of data.

Learning a target BMRS-Tree follows the same procedure as vanilla Decision Trees, using CART after arranging the attribute table. However, its evaluation differs significantly. Traditional Decision Tree evaluation focuses on cross-validation to prevent overfitting. In contrast, BMRS-Tree evaluation focuses on the purity of leaf nodes. To fit phonological data, CART minimizes Entropy in the leaf nodes (Shannon, 1948). **In the case of non-variable mappings, we propose that all leaf nodes in a well-fitted BMRS-Tree must achieve zero Entropy**, i.e. they are 100% pure. The reasons are as follows:[8]

1. BMRS-Tree learning aims to reconstruct deterministic phonological rules, rather than to generalize over unseen data (test set). In the two case studies discussed in Sections 3.2 and 4.2, all possible data are provided as the training set.[9] Thus, the training data should not be treated as samples from a larger distribution, but as a complete representation of the rule-governed system. The learning task then requires the model to fully capture and account for all observed patterns.

2. Non-variable mappings require each input to correspond to exactly one output, i.e., no free variation or probabilistic choice. If a leaf node contains multiple output classes, it introduces ambiguity, implying that a single context could trigger more than one realization. This contradicts the nature of non-variability and obstructs the derivation of a clear, well-defined rule. Zero entropy ensures that each decision path leads to a unique and unambiguous output—one that is interpretable, consistent, and faithful to the phonological data.

3. In traditional machine learning, overfitting refers to a model capturing too many exceptions or "outliers," reducing its ability to generalize. In contrast, exceptions in phonology are integral to the language and must be explicitly modeled; they are not noise to be ignored. Thus, requiring all leaf nodes to be 100% pure doesn't lead to overgeneration; rather, it helps prevent it. BMRS naturally handles exceptions through structured exception-filtering logic, represented using a series of embedded "if $exception_1(x)$ then $path_1$ else $path_2$" expressions.

In summary, the BMRS-Tree's uniqueness lies in its 100% accurate fit: its goal is to reconstruct the system rather than generalize from partial data. For interpretability, the BMRS-Tree can be validated against real phonological data, deriving rules and constraints from it (see Section 3.2 for a case study), which could be compared with already observed patterns.

3.2 Case Study 1: High Tone Shift in Kibondei

In our toy grammar, which is loosely based on the high tone shift patterns observed in Kibondei

[8] We will leave open the question of variability for the future.

[9] In Section 3.2, all training data have string lengths ranging from 1 to 8. However, we propose that the BMRS-Tree learned from the training set can also successfully generalize to strings longer than 9, due to the use of Global Feature Predicates, which are distance-insensitive.

(Merlevede, 1995; Lamont, 2024), elements can be **high-toned** (denoted by H), **low-toned** (L), or **unspecified** for tone (0). For simplicity, it is assumed that no more than one high-toned element is present in the input. The grammar operates under the following hypothetical rules:

Rule 1: L in the UR faithfully surfaces ($L \rightarrow L$; $L \nrightarrow H$, $L \nrightarrow 0$).

Rule 2: H shifts to the penultimate element if possible (e.g., $H000 \rightarrow 00H0$). It can only replace 0 and leaves the original position in 0.

Rule 3: H cannot shift across L. If an L intervenes between the H and the penultimate element, then H shifts only up to the L (e.g., $H000L000 \rightarrow 000HL000$; $H000L000 \nrightarrow 0000L0H0$).

Rule 4: H cannot surface on the final element. Underlyingly final H shifts to the penultimate position if possible (e.g., $000H \rightarrow 00H0$), and deletes if the penultimate position is occupied by an L (e.g., $00LH \rightarrow 00L0$; $00LH \nrightarrow 0HL0$).

To demonstrate the learning results of BMRS-Trees, we generated a dataset of UR-SF pairs, with each string having a length between 1 and 8, sufficient to capture potential long-distance dependencies in the high tone shift. The algorithm used to generate the dataset is provided in pseudocode in Appendix C. Ten representative data samples are presented in Table 3:

Data	UR	SF	Data	UR	SF
1	H00L	00HL	6	000H00L	00000HL
2	LH000	L00H0	7	L0000HL	L0000HL
3	000L0H	000LH0	8	00H0000L	000000HL
4	LH0L00	L0HL00	9	L0H000L0	L0000HL0
5	LH000L0	L000HL0	10	L0H0000L	L00000HL

Table 3: Data Samples of Kibondei High Tone Shift

The first step in attribute aggregation is to enumerate each symbol $\sigma \in \Sigma$: $\Sigma = \{H, L, 0\}$.

Next, by aggregating feature predicates from each index within each data (see Section 3.1), we obtain the attribute table for learning the BMRS-Tree of the target feature predicate $H_o(x)$. Running CART on this table (see Section 2.2) with scikit-learn generates the Tree diagram of $H_o(x)$, displayed in Figure 3.

The BMRS-Tree begins at the root node $L(x)$, which checks for the presence of L at the current index. If $L(x) = \top$, Rule 1 ensures that H cannot occur at the same index, returning $\bot$ for $H_o(x)$.

Continuing down, the BMRS-Tree evaluates whether the current index is valid to receive H shift: Rule 3 ensures that an L at the succeeding index ($L(s(x)) = \top$) blocks this shift, allowing the current index to be a valid alternative; Rule 2 prefers high tone shifts to the penultimate position, which is evaluated by $\_(s^2(x))$.[10]

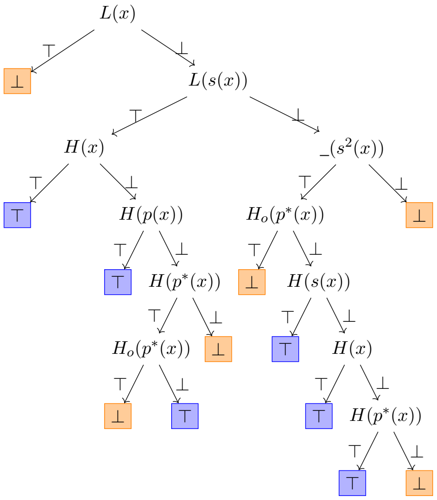


Figure 3: BMRS-Tree $H_o(x)$

Rule 3 also implicitly ascertains that the closest H can successfully shift to the current index without encountering an intervening L:

When the immediate successor is L ($L(s(x)) = \top$), then the BMRS-Tree returns $\top$ if an H exists either at the current index ($H(x) = \top$) or the immediately preceding index ($H(p(x)) = \top$). The challenge arises when locating H among all predecessors, evaluated by $H(p^*(x))$. If $H(p^*(x)) = \top$, it confirms an H at some index to the left but doesn't verify if it's blocked by an L. A common solution is to recursively test two competing elements H and L to decide which one appears earlier when looking ahead backward, using a manually-formulated function defined as:

$$
\begin{aligned}
Hprec(x) &= \text{if } H(p(x)) \text{ then } \top \text{ else} \\
&\qquad \text{if } L(p(x)) \text{ then } \bot \text{ else } Hprec(p(x))
\end{aligned}
$$

This function finds the closest H or L backward from the current index, successfully indicating whether an H can shift without interruption.

By comparison, learned from real dataset, the BMRS-Tree introduces a refined method by using an output-dependent feature predicate $H_o(p^*(x))$, which checks if an H has been output among all predecessors. If $H(p^*(x)) = \top$ and $H_o(p^*(x)) = \bot$, it confirms that this H can shift to the current index, simplifying the decision-making process without recursive backtracking.

[10] See Footnote 7 for discussion.

When the current position is penultimate (_$(s^2(x)) = \top$), the BMRS-Tree also ascertains that $H_o(p^*(x)) = \top$ for an uninterrupted H shift, validating subsequent paths: $H(s(x))$ follows Rule 4 for final H shifts, and $H(x)$ and $H(p^*(x))$ check for H at the current index or among predecessors.

In general, the BMRS-Tree intricately captures interactions of H and L in an artificial dataset by integrating output-dependent feature predicates, simplifying and optimizing the process of locating valid H shifts. This approach enhances its capability of handling wider range of phonological transformations (Oakden, 2021).

4 BMRS-Net

4.1 Implementation

Using phonological features as embeddings allows parallel processing of multiple BMRS-Trees, forming a complex network-like structure, referred to as BMRS-Net. This paper proposes a method of vectorization that treats phonological features as Boolean values (cf. Prickett, 2021).

Central to this method is the redefinition of each symbol $\sigma \in \Sigma$. Traditionally seen as mere symbols (characters) in strings, in the BMRS-Net symbols in Σ are understood as underlying components (i.e. phonological features) of each segment. Thus, Σ can be defined as:

$$\Sigma = \{[F_1], [F_2], \ldots, [F_m]\}$$

where each $[F_i]$ represents a Boolean phonological feature that returns either $\top$ (1) or $\bot$ (0); m equals $|\Sigma|$, the size of the Symbol Set, which signifies the total count of unique features.

Each segment ω from the vocabulary V (also referred to as the phoneme inventory) is then represented as an $(m+1)$-dimensional vector. This vector is constructed by assessing each phonological feature $[F_i]$ for ω, plus an extra 0 as the final element, which represents an additional phonological feature specifically for the boundary symbol _. This boundary feature returns $\top$ only when evaluated on the boundary symbol _. Formally, a character ω in V can be represented by the vector:

$$\vec{v} = ([F_1](\omega), [F_2](\omega), \ldots, [F_m](\omega), 0)$$

We introduce the embedding matrix E, a one-to-one mapping from each segment to its vector representation, expressed as:

$$E : \omega \rightarrow \vec{v}$$

The notation $E(\omega) = \vec{v}$ denotes the vector associated with a segment ω, and its inverse function $E^{-1}(\vec{v}) = \omega$ denotes the retrieval of the original segment from its vector representation.

By definition, BMRS-Net is the parallel connection of $m + 1$ BMRS-Trees, where $m = |\Sigma|$.

The embedding matrix E facilitates transformation of phonological data into a vector format, essential for BMRS-Net processing. It computes the output vector $\vec{v}_o$ for a given input segment ω. The input vector $\vec{v}$ with respect to the input segment ω and the corresponding output vector $\vec{v}_o$ are respectively defined as:

$$\vec{v} = E(\omega)$$

$$\vec{v}_o = ([F_1]_o(\omega), [F_2]_o(\omega), \ldots, [F_m]_o(\omega), 0)$$

As can be noticed, $\vec{v}_o$ includes the same features as $\vec{v}$, with an additional zero for the boundary symbol _. Once $\vec{v}_o$ is computed, the inverse function of E is employed to retrieve the corresponding segment for further analysis or processing. The retrieved segment, denoted as ω_o, is obtained through:

$$\omega_o = E^{-1}(\vec{v}_o)$$

Figure 4 visualizes the BMRS-Net transformation of a given index i (each grey block denotes an individual target feature predicate):

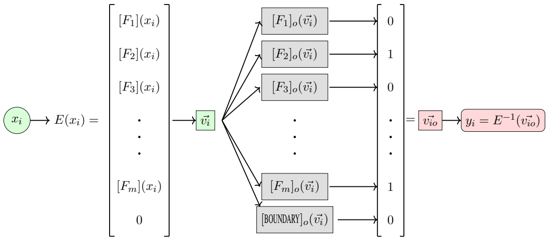


Figure 4: BMRS-Net[11]

4.2 Case Study 2: Rhotacization in Mandarin

This phenomenon refers to the transformation of a non-rhotic sound into a rhotic one, typically resembling a [ɻ]-like sound (Chao, 1968; Lu, 1995; Eckert, 2018). It generally occurs at the syllabic level, and adding the suffix -ɚ induces alternations within the rhyme.

The dataset, summarized in Table 4, draws from research by Lin (1989), Duanmu (2007), and Zhu

[11] For simplification, $[F]_o(\vec{v})$ denotes the same as $[F]_o(E^{-1}(\vec{v}))$.

(2023). This training set includes only the rhyme components (nucleus + coda) of stems plus the suffix -ə˞. Glide components in the onset parts of URs are also included if they trigger Mid Vowel Alternation (discussed later in this section); rhotacization can also alter some segments into glides in SFs.

UR	SF	UR	SF	UR	SF	UR	SF
i-ə˞	jə˞	u-ə˞	u˞	ə-ə˞	ɤ˞	əi-ə˞	ə˞
in-ə˞	jə˞	un-ə˞	u˞	jə-ə˞	je˞	ai-ə˞	a˞
iŋ-ə˞	jə̃˞	uŋ-ə˞	ũ˞	ɥə-ə˞	ɥe˞	əu-ə˞	ou˞
y-ə˞	ɥə˞	a-ə˞	a˞	wə-ə˞	wo˞	au-ə˞	au˞
yn-ə˞	ɥə˞	an-ə˞	a˞	ən-ə˞	ə˞		
yŋ-ə˞	ɥə̃˞	aŋ-ə˞	ã˞	əŋ-ə˞	ə̃˞	ɨ-ə˞[12]	ə˞

Table 4: Mandarin Rhotacization Dataset

Observing the dataset, we can make several generalizations, some consistent with Zhu (2023):

1. **Alveolar nasal coda [n]** does not nasalize the surrounding vowel, while **velar nasal [ŋ]** does. Both nasal codas are deleted in SFs.

2. The **segment undergoing rhotacization** in the SF varies significantly depending on the **nuclei of the stems** in URs. When the stem nucleus is:

- **High front vowels [i]/[y]** (Column 1 Table 4): [i] and [y] reduce to glides [j] and [ɥ], with [ə˞] becoming the nucleus in SF; the suffix vowel [ə˞] becomes the nucleus in the SF.

- **Back or low vowel [u]/[a]** (Column 2 Table 4): [u] and [a] remain as the nucleus and undergo rhotacization; the suffix vowel [ə˞] then deletes.

- **Mid vowel [ə]** (Column 3 Table 4): [ə] firstly undergoes Mid Vowel Alternation, summarized in Table 5, then the altered vowel becomes the nucleus and undergoes rhotacization; the suffix vowel [ə˞] deletes.

- **Diphthong** (Column 4 Table 4): The coda vowel [i] deletes, and the "real" nucleus undergoes rhotacization. The coda vowel [u] undergoes rhotacization while the preceding vowel remains unchanged or undergoes Mid Vowel Alternation (ə→ o / __ u). In both scenarios, the suffix vowel [ə˞] deletes.

- **High central vowel [ɨ]** (the last line of Column 4 Table 4): [ɨ] is assumed to undergo rhotacization but surfaces as [ə˞], with the suffix vowel [ə˞] being deleted.

Description	Rule
Undergoes [+front] assimilation	ə → e / {j, ɥ} __
Undergoes [+back] assimilation	ə → o / {w __, __ u}
Surfaces as [ɤ] in open syllable stems	ə → ɤ / __ $]_\sigma$
Remains unchanged with nasal coda	ə → ə / __ {n, ŋ}

Table 5: Mid Vowel Alternations

Given that some segments are deleted in the training set (Table 4), we propose inserting the symbol 0, representing a zero vector where every output feature predicate returns $\bot$, to indicate deleted elements in the output.[13] This alignment ensures that each UR-SF pair is of the same length, consistent with BMRS' index-by-index nature. The aligned training set is presented in Table 6:

UR	SF	UR	SF	UR	SF	UR	SF
i ə˞	j ə˞	u ə˞	u˞ 0	ə ə˞	ɤ˞ 0	ə i ə˞	ə˞ 0 0
i n ə˞	j 0 ə˞	u n ə˞	u˞ 0 0	j ə ə˞	j e˞ 0	a i ə˞	a˞ 0 0
i ŋ ə˞	j 0 ə̃˞	u ŋ ə˞	ũ˞ 0 0	ɥ ə ə˞	ɥ e˞ 0	ə u ə˞	o u˞ 0
y ə˞	ɥ ə˞	a ə˞	a˞ 0	w ə ə˞	w o˞ 0	a u ə˞	a u˞ 0
y n ə˞	ɥ 0 ə˞	a n ə˞	a˞ 0 0	ə n ə˞	ə˞ 0 0		
y ŋ ə˞	ɥ 0 ə̃˞	a ŋ ə˞	ã˞ 0 0	ə ŋ ə˞	ə̃˞ 0 0	ɨ ə˞	ə˞ 0

Table 6: Mandarin Rhotacization Data (after alignment)

For Σ, we refer to the Feature Charts from Hayes (2009) to select relevant phonological features. For vowels, we include attributes like **[high]**, **[low]**, **[front]**, **[back]**, and **[round]**. For the three glides observed ([j], [ɥ] and [w]), we use **[cons]** and **[syll]**. The **[nasal]** feature covers nasalized vowels and two nasal codas ([n] and [ŋ]), and additional features **[COR]** and **[DOR]** help distinguish them. **[rhotic]** is specifically used for rhotacized vowels.[14]

Σ contains all the features above plus the [BOUNDARY] feature; and the Embedding Matrix E is outlined using this subpart of the Feature Chart (see Appendix D).

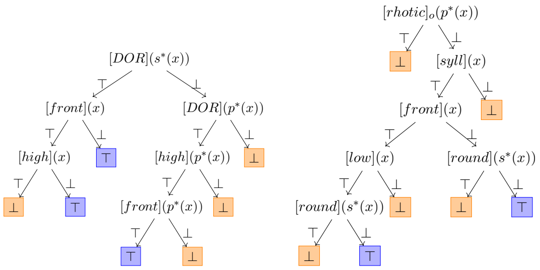
Following the established procedures from Sections 3.1 and 4.1, we can learn all the target feature predicates on Σ, provided in Appendix E. Some representative Tree diagrams will be reproduced in the following discussion for illustration:

1. **Nasal Assimilation** is controlled by $[nasal]_o(x)$ (Figure 5a), and is only triggered by a surrounding [ŋ], represented by [+DOR].

[12] Two syllabic fricatives ([z̩] and [ʐ̩]), also called apical or fricative vowels, or syllabic approximants (cf. Lee-Kim, 2014) are merged into the high central unrounded vowel [ɨ], according to (Cheng, 1973) and by convention.

[13] Similar to early OT methods where unparsed segments were considered deleted.

[14] The Hayes (2009) feature for rhotacization is [+COR, +anterior, +distributed, –strident]. Here for simplicity, we use the informal feature [rhotic].



(a) $[nasal]_o(x)$ (b) $[rhotic]_o(x)$

Figure 5

When $[DOR](s^*(x)) = \top$ ([ŋ] appears among successors), then all the vowels except [+high, +front] will be nasalized in the SF, reflected in the data: [uŋ-ɚ] → [ũ˞], [aŋ-ɚ] → [ã˞], [əŋ-ɚ] → [ɚ̃].

When $[DOR](p^*(x)) = \top$ ([ŋ] precedes the current index), the output segment will be nasalized only if there is [+high, +front] among its predecessors, reflected in: [iŋ-ɚ] → [jɚ̃], [yŋ-ɚ] → [ɥɚ̃].

2. **Rhotacization** is controlled by $[rhotic]_o(x)$ (Figure 5b), which decides whether the current segment can undergo rhotacization in the SF (i.e. to receive the [rhotic] feature).

The root node $[rhotic]_o(p^*(x))$ checks whether [+rhotic] has surfaced before the current segment. As observed from Table 4, [+rhotic] must be aligned to the final segment and surface at the final position, saying that $[rhotic]_o(p^*(x))$ actually checks whether the output string has reached the end: if it has ($[rhotic]_o(p^*(x))$ returns $\top$), then every segment from the current position will delete.

In the rest of Figure 5b, all leaf nodes returning $\top$ appear when $[round](s^*(x))$ returns $\perp$, which imposes a constraint-like condition on that a vowel cannot receive [rhotic] if it's followed by a [+round] element (in Mandarin, [u]).[15] This is also coherent to the dataset: if a vowel is succeeded by a [u], then [u] is always the one to receive [rhotic].

There are only two leaf nodes returning $\top$ (the blue nodes), which denote respectively:

- **[+front, -low]**: **[a]** (the bottom-left $\top$);
- **[-front]**: **[ɨ]**, **[ə]** and **[u]** (the bottom-right $\top$).

3. **Glide Formation** ([j], [ɥ]) in SFs offers an explanation for why [+high, +front] cannot be rhotacized. This glide formation is controlled by $[syll]_o(x)$ (Figure 6a). It also starts with $[rhotic]_o(p^*(x))$, restricting that the output string (SF) has not yet reached the end. And $[syll](x)$ asserts that [-syll] segments won't surface as [+syll]. The rest of the two intermediate nodes $[front](x)$ and $[high](x)$ denotes respectively two categories of vowels that remain [+syll] in the SF:

- **[-front]**: [ə] and [u];
- **[+front, -high]**: [a].

The bottom-left $\perp$ leaf node (in orange) denotes exactly the category that will possibly be altered to glides (or even deleted): **[+front, +high]**, consistent with the data in Column 1 Table 4).

4. **Mid Vowel Alternation** is applied to ə in the stem before deciding whether it receives [rhotic] or not. It is controlled by three predicates: $[front]_o(x)$, $[back]_o(x)$, and $[round]_o(x)$ (refer to Table 9). After being applied to the underlying ə, they are reproduced in Figures 6b, 6c and 6d.

All three BMRS-Trees start with $[syll](p^*(ə))$, checking whether a [+syll] segment (i.e. vowel) precedes ə. This presents a restriction that ə alternates only if it's the stem's nucleus or the so-called "real" nucleus of a diphthong; the ə in the suffix -ɚ or as the coda vowel doesn't alternate (though it never appears as the coda in Mandarin).

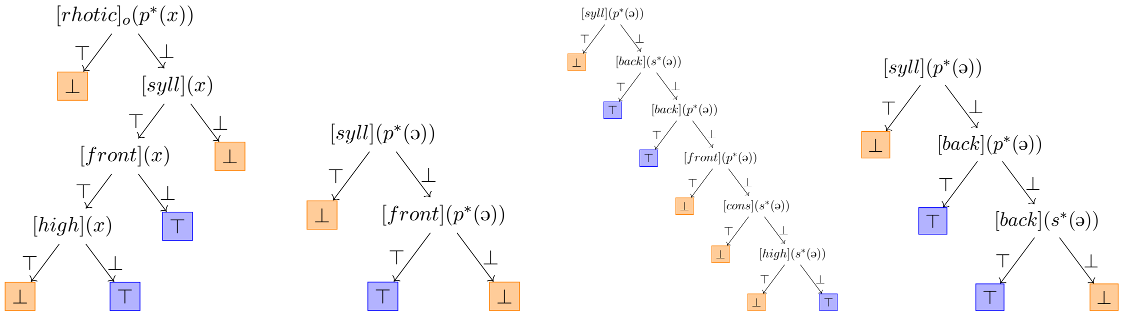
Ascertaining that ə appears as the stem's nucleus ($[syll](p^*(ə)) = \perp$, this continues as a prerequisite in the following discussion), Figure 6b then successfully models **[+front] assimilation**: a [+front] segment preceding the ə assimilates it into a front vowel [e] ($[front](p^*(ə)) = \top$).

The two upper $\top$ nodes in Figure 6c model **[+back] assimilation**: ə is [+back] assimilated when there exists a [+back] segment before or after it. Continuing down, $[front](p^*(ə))$ filters out two front glides ([j] and [ɥ]) that license the [+front] assimilation; $[cons](s^*(ə))$ filters out two cases where ə is followed by nasal codas [n] and [ŋ] – the only two consonants existent in our dataset ([ən-ɚ] → [ɚ], [əŋ-ɚ] → [ɚ̃]).

$[high](s^*(ə))$ models another possibility: **pseudo-[+back] assimilation**, when followed by a [+high] segment. This is consistent with the piece of data: [ə-ɚ] → [ɤ˞], in comparison with [əi-ɚ] → [ɚ] (the ə followed by [i] doesn't alternate). In fact, this is also consistent with Line 3 Table 5 that ə surfaces as ɤ in the open syllable stem (cf. Duanmu, 2007).

Figure 6d is almost identical to the upper part of 6c, both seeking a [+back] environment. To generalize, ə automatically receives [+round] when it re-

[15] In Mandarin, [y] never appears in complex nuclei or diphthongs.



(a) $[syll]_o(x)$ (b) $[front]_o(ə)$ (c) $[back]_o(ə)$ (d) $[round]_o(ə)$

Figure 6

ceives [+back] from its surrounding context, which is to say, **[+back] triggered by "real" [+back] assimilation innately carries [+round]**.

All the discussion above serves as an illustration of complete transparency and interpretability of BMRS-Trees learned via CART. Collectively, BMRS-Net successfully fitted the dataset and is capable of efficiently performing complex string (vector) transformations.

5 Future Research Directions

First, regarding the class of string functions, the High Tone Shift in Kibondei can be modeled by a Subsequential function (Heinz and Lai, 2013; Heinz, 2018), while the Rhotacization in Mandarin could be considered Output-Strictly local (Chandlee et al., 2015), at least in this paper, due to its dependence on the previous output to determine transformations. However, in our implementation, all categories of feature predicates (including symbolic, local, global, and output-dependent) were aggregated to form the attributes used in Decision Trees for phonological analysis (Section 3.1). Therefore, it would be beneficial to systematically analyze which categories of feature predicates are sufficient to model different string function classes.

Second, the unequal string lengths for Mandarin Rhotacization (Section 4.2) is handled a little unusually in this paper, and the use of zero vector (0) to indicate deleted segments is obviously not scalable to inserted one. Thus how to extend the current implementation to handle both deletion and epenthesis remains open for further exploration. According to a suggestion from an anonymous reviewer, the use of licensing functions and copy sets, as discussed in the work of Courcelle and Engelfriet (2012), offers a promising direction. Besides, the integration of order-preserving functions (as explored by Lindell and Chandlee, 2016) could also enable deriving both deletion and epenthesis.

If the successful categorization of feature predicates for different string function classes is achievable, and handling epenthesis becomes feasible, then our BMRS implementation could serve as a versatile tool for analyzing various classes of string functions and a broader range of phonological transformations, with enhanced flexibility and expressivity.

6 Conclusion

This paper presents the implementation of *BMRS-Trees* and *BMRS-Net* as an automated BMRS predicate learner, requiring only minimal human input, i.e., symbol (or feature) selection. Their successful application to two non-trivial (yet still limited) phonological phenomena substantiates their potential as an automation tool for researching phonological transductions, from segmental alternation, deletion to long-distance shifts (with epenthesis left for future exploration). The results offer a promising alternative to traditional rule- or constraint-based approaches, advancing the integration of machine learning in computational phonology.

Acknowledgments

This work is based on the author's MA dissertation at University College London. I am extremely grateful to Andrew Lamont for introducing BMRS to me, suggesting possible research directions and reviewing my draft, and to Sebastian Schuster for discussing and inspiring me about correlations between BMRS and Decision Trees and reviewing my draft. I also thank the anonymous reviewers for their valuable comments and suggestions, which greatly improved the quality of this paper. All remaining errors are my own.

References

Siddharth Bhaskar, Jane Chandlee, Adam Jardine, and Christopher Oakden. 2020. Boolean monadic recursive schemes as a logical characterization of the subsequential functions. In *Language and Automata Theory and Applications - LATA 2020*, Lecture Notes in Computer Science, pages 157–169. Springer.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. 1984. *Classification and Regression Trees*. Wadsworth International Group.

Jeroen Breteler. 2017. Deriving bounded tone with layered feet in harmonic serialism: The case of saghala. *Glossa: a journal of general linguistics*, 2(1).

Jane Chandlee. 2014. *Strictly Local Phonological Processes*. Ph.d. thesis, University of Delaware.

Jane Chandlee. 2023. Decision trees, entropy, and the contrastive feature hierarchy. *Proceedings of the Linguistic Society of America*, 8(1):5465.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2015. Output strictly local functions. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, pages 112–125, Chicago, USA. Association for Computational Linguistics.

Jane Chandlee, Jeffrey Heinz, and Adam Jardine. 2018. Input Strictly Local opaque maps. *Phonology*, 35:1–35.

Jane Chandlee and Adam Jardine. 2014. Learning phonological mappings by learning Strictly Local functions. In *Proceedings of the 2013 Meeting on Phonology (UMass Amherst)*, Proceedings of the Annual Meetings on Phonology. LSA.

Jane Chandlee and Adam Jardine. 2021. Computational universals in linguistic theory: Using recursive programs for phonological analysis. *Language*, 93:485–519.

Yuanren Chao. 1968. *A Grammar of Spoken Chinese*. University of California Press, Berkeley, California.

Chin-Chuan Cheng. 1973. *A Synchronic Phonology of Mandarin Chinese*. De Gruyter Mouton, Berlin, New York.

Noam Chomsky and Morris Halle. 1968. The sound pattern of english.

Bruno Courcelle and Joost Engelfriet. 2012. *Graph Structure and Monadic Second-Order Logic, a Language Theoretic Approach*. Cambridge University Press.

San Duanmu. 2007. *The Phonology of Standard Chinese*, 2nd edition. Oxford University Press Inc., New York.

Penelope Eckert. 2018. *Meaning and Linguistic Variation: The Third Wave in Sociolinguistics*. Cambridge University Press.

Aurelien Geron. 2019. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd edition. O'Reilly Media, Inc.

Bruce Hayes. 2009. *Introductory Phonology*. Wiley-Blackwell Publication, UK.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. In Larry Hyman and Frans Plank, editors, *Phonological Typology*, Phonetics and Phonology, chapter 5, pages 126–195. De Gruyter Mouton.

Jeffrey Heinz and Regine Lai. 2013. Vowel harmony and subsequentiality. In *Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13)*, pages 52–63, Sofia, Bulgaria.

Wenyue Hua, Huteng Dai, and Adam Jardine. 2021. Learning underlying representations and input-strictly-local functions. In *Proceedings of the 37th West Coast Conference on Formal Linguistics*, pages 143–151. Cascadilla Proceedings Project.

Adam Jardine and Christopher Oakden. 2023. Computing Process-Specific Constraints. *Linguistic Inquiry*, pages 1–9.

Andrew Lamont. 2024. Shift is derived. *Journal of Linguistics*.

Sang-Im Lee-Kim. 2014. Revisiting mandarin 'apical vowels': An articulatory and acoustic study. *Journal of the International Phonetic Association*, 44(3):261–282.

Yen-Hwei Lin. 1989. *Autosegmental Treatment of Segmental Processes in Chinese Phonology*. Phd dissertation, The University of Texas at Austin, Austin.

Steven Lindell and Jane Chandlee. 2016. A logical characterization of input strictly local functions. Presented at the Fourth Workshop on Natural Language and Computer Science, in affiliation with LICS 2016.

Yunzhong Lu. 1995. *Putonghua de Qingsheng he Erhua [Neutral Tones and Rhotacization in Mandarin]*. Shangwuyinshuguan, Beijing, China.

Andrea Merlevede. 1995. Een schets van de fonologie en morfologie van het bondei. Ma thesis, Leiden University, Leiden. Written in Dutch.

Christopher Donal Oakden. 2021. *Modeling Phonological Interactions Using Recursive Schemes*. Ph.d. dissertation, Rutgers University, School of Graduate Studies, New Jersey. Degree granted in October 2021.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Brandon Prickett. 2021. *Learning Phonology With Sequence-To-Sequence Neural Networks*. Ph.d. dissertation, University of Massachusetts Amherst.

Alan S. Prince and Paul Smolensky. 2002. Optimality theory: Constraint interaction in generative grammar. Technical Documentation 991031549929204646, Rutgers University, Rutgers University. Essentially identical to the Tech Report (July 1993), with new pagination but the same footnote and example numbering; corrections of typos, oversights, and outright errors; improved typography; and occasional small-scale clarificatory rewordings.

J. Ross Quinlan. 1986. Induction of decision trees. *Machine Learning*, 1:81–106.

C. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Ziling Zhu. 2023. Modeling mandarin rhotacization with recursive schemes. *Proceedings of the Linguistic Society of America*, 8(1):5510.

A Modeling Tonal Shift and Spread with BMRS

Below is a brief example of BMRS transduction on a tonal system.

Saghala (Breteler, 2017) has a tone system that contrasts only high-toned elements (denoted by H) with unspecified ones (0). An underlying H shifts to the next position and then spreads one position further to the right (e.g., $00 \rightarrow 00$, $H00 \rightarrow 0HH$, $H000 \rightarrow 0HH0$, $0H00 \rightarrow 00HH$).

Assuming $\Sigma = H, 0$, we can define $H_o(x)$ to check whether the current index outputs H:

$$H_o(x) = \text{if } H(x) \text{ then } \bot \text{ else (if } H(p(x)) \text{ then } \top \text{ else } H(p^2(x)) \text{)}$$

This captures the rightward shift-and-spread behavior of H. The tree diagram in Figure 7 also visualizes this same behavior:

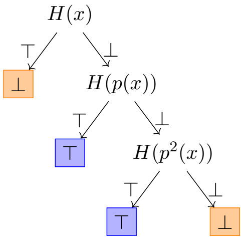


Figure 7: $H_o(x)$

Table 7 illustrates the index-by-index transductions on two input-output mappings:

	0	1	2	3	4	0	1	2	3	4
input	_	H	0	0	0	_	0	H	0	0
$H(x)$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$
$H(p(x))$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
$H(p^2(x))$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$
$H_o(x)$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\top$
output	_	0	H	H	0	_	0	0	H	H

Table 7: $H000 \rightarrow 0HH0, 0H00 \rightarrow 00HH$

B Classification and Regression Tree (CART) Algorithm

CART (Breiman et al., 1984) builds binary trees. When used for classification, CART aims to split the data into subsets that are as "homogeneous" (pure) as possible with respect to the target attribute.

Entropy, borrowed from Information Theory (Shannon, 1948), is a common metric to quantify the degree of homogeneity or impurity in a dataset, and is employed as the split criterion in this paper. For a binary classification task that returns Boolean values, Entropy H of a dataset D is defined as:

$$H(D) = -p_0 \log_2(p_0) - p_1 \log_2(p_1)$$

where p_0 and p_1 refer respectively to the proportions of instances returning $\bot$ in the dataset and to that of instances returning $\top$. Entropy $H(D)$ reaches its maximum when $\top$ instances and $\bot$ instances are equally distributed (the dataset D being the most "heterogeneous" or impure) and its minimum (zero) when the dataset contains only one class (completely pure).

CART grows a Decision Tree in these steps:

1. **Calculate Initial Entropy**: The algorithm begins by calculating the Entropy of the entire dataset $H(D)$, which gives a baseline measure of impurity.

2. **Evaluate Each Attribute and Choose the Best Split**: For each attribute, CART firstly considers its split and calculates the Entropy of two resulting subsets. It then computes the **Information Gain**, which is the reduction in Entropy from the initial dataset to the combination of its two subset. The Information Gain IG from splitting dataset D on attribute A is defined as:

$$IG(D, A) = H(D) - \left(\frac{|D_0|}{|D|} H(D_0) + \frac{|D_1|}{|D|} H(D_1)\right)$$

where D_0 and D_1 denote the subsets formed by the split on attribute A, and $|D_0|$, $|D_1|$

and $|D|$ denote respectively the number of instances in the corresponding set.

The attribute that yields the largest Information Gain is selected for that split.

3. **Split the Subsets Recursively**: The process of splitting based on Information Gain continues recursively for each subset, creating decision nodes and branches, until all instances in a subset belong to the same class, or no further information gain can be achieved (because all the attributes are used up).

4. **Assign Leaf Nodes**: When no further splits are possible or necessary, the remaining data in each terminal node is assigned a label based on the majority class within that subset, forming a leaf node.

Focusing on reducing uncertainty at each step, CART constructs Decision Trees that classify the dataset as accurately as possible, while being relatively easy to interpret and to visualize using the scikit-learn library.

C Algorithm to Generate the Dataset for High Tone Shift in Kibondei

Algorithm 1 Generate Input Strings

```
Require: $min\_len$, $max\_len$
Ensure: A list of input strings consisting of $H$ (at most 1), $L$, and 0
$inputs \leftarrow [\,]$
for $length \leftarrow min\_len$ to $max\_len$ do
    $strings \leftarrow$ all combinations of $L$ and 0 of $length$
    for all $s \in strings$ do
        Append $s$ to $inputs$
        for $i \leftarrow 0$ to $length(s) - 1$ do
            $modified \leftarrow s$ with character at position $i$ replaced by $H$
            Append $modified$ to $inputs$
        end for
    end for
end for
return $inputs$
```

For reference, when $min\_len = 1$ and $max\_len = 8$, Algorithm 1 returns a list of length 4096, i.e., containing 4096 possible inputs.

Algorithm 2 Map Input to Output

```
Require: A string $input$
Ensure: The output string after applying the BMRS transduction
if $input$ ends with $0H$ then
    Replace the suffix $0H$ with $H0$
end if
if $input$ ends with $LH$ then
    Replace the suffix $LH$ with $L0$
end if
Replace every substring matching pattern H(0*)L with:
     same number of 0's as in the match, followed by $HL$
if $input$ ends with a substring matching pattern H0+ then
    Replace it with:
        one fewer 0 followed by $H0$
end if
return $input$
```

D Mandarin Feature Chart

Phonological features selected for Section 4.2 are presented in Table 8.[16]

	[high]	[low]	[front]	[back]	[round]	[cons]	[syll]	[nasal]	[COR]	[DOR]	[rhotic]	[BOUNDARY]
i	1	0	1	0	0	0	1	0	0	0	0	0
y	1	0	1	0	1	0	1	0	0	0	0	0
ɨ	1	0	0	0	0	0	1	0	0	0	0	0
u	1	0	0	1	1	0	1	0	0	0	0	0
u˞	1	0	0	1	1	0	1	0	0	0	1	0
ũ˞	1	0	0	1	1	0	1	1	0	0	1	0
e	0	0	1	0	0	0	1	0	0	0	0	0
e˞	0	0	1	0	0	0	1	0	0	0	1	0
ə	0	0	0	0	0	0	1	0	0	0	0	0
ə˞	0	0	0	0	0	0	1	0	0	0	1	0
ə̃˞	0	0	0	0	0	0	1	1	0	0	1	0
ɤ	0	0	0	1	0	0	1	0	0	0	0	0
ɤ˞	0	0	0	1	0	0	1	0	0	0	1	0
o	0	0	0	1	1	0	1	0	0	0	0	0
o˞	0	0	0	1	1	0	1	0	0	0	1	0
a	0	1	1	0	0	0	1	0	0	0	0	0
a˞	0	1	1	0	0	0	1	0	0	0	1	0
ã˞	0	1	1	0	0	0	1	1	0	0	1	0
j	1	0	1	0	0	0	0	0	0	0	0	0
ɥ	1	0	1	0	1	0	0	0	0	0	0	0
w	1	0	0	1	1	0	0	0	0	0	0	0
n	0	0	0	0	0	1	0	1	1	0	0	0
ŋ	0	0	0	0	0	1	0	1	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
_	0	0	0	0	0	0	0	0	0	0	0	1

Table 8: Embedding Matrix

E Mandarin Rhotacization BMRS-Trees

In this appendix, both original Decision Tree diagrams generated directly by scikit-learn and their simplified (remade) versions are presented in Table 9.[17]

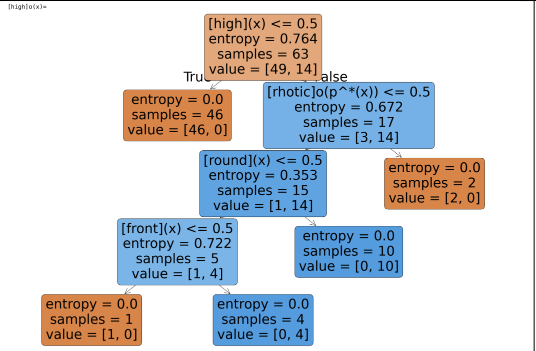
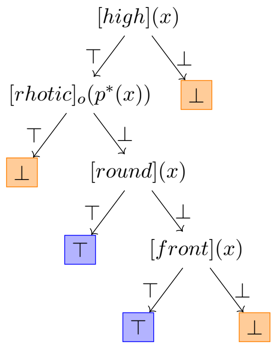
[16] The feature [BOUNDARY] and the default boundary symbol _ are also included in E, for the sake of completeness.

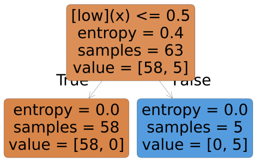
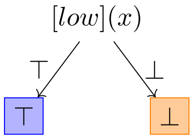
[17] As can be noticed from Table 4, all [+cons] segments (i.e., [n] and [ŋ]) are deleted and don't surface in the output.

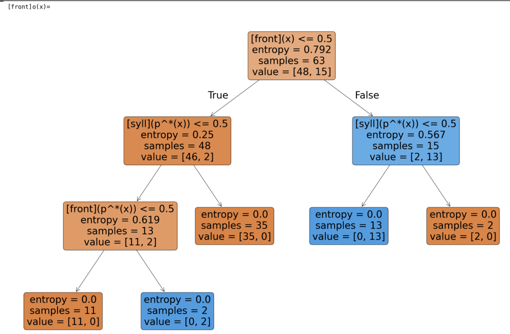
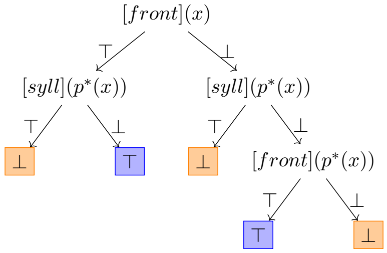
The original BMRS-Tree diagrams generated by scikit-learn can appear perplexing due to its exceedingly detailed node information, and somewhat counter-intuitive as each node tests whether a feature predicate returns "False": the node checks whether the truth value is $<= 0.5$. Therefore, original and remade versions look like horizontal mirror images of each other. And given the low readability, a remade version is reproduced below in Table 9 and used in the main body of this paper for better visualization.

One prominently essential parameter exclusively existing here in Column "Original", Table 9 is **Entropy**: all terminal leaf nodes' Entropy equals zero, which is a key indication of 100% accurate fit.

BMRS-Tree	Original	Reproduced

$[high]_o(x)$	 [high]o(x)=
	[high](x) <= 0.5
	entropy = 0.764
	samples = 63
	value = [49, 14]
	True
	False
	entropy = 0.0
	samples = 46
	value = [46, 0]
	[rhotic]o(p^*(x)) <= 0.5
	entropy = 0.672
	samples = 17
	value = [3, 14]
	[round](x) <= 0.5
	entropy = 0.353
	samples = 15
	value = [1, 14]
	entropy = 0.0
	samples = 2
	value = [2, 0]
	[front](x) <= 0.5
	entropy = 0.722
	samples = 5
	value = [1, 4]
	entropy = 0.0
	samples = 10
	value = [0, 10]
	entropy = 0.0
	samples = 1
	value = [1, 0]
	entropy = 0.0
	samples = 4
	value = [0, 4]
		 $[high](x)$
		⊤
		⊥
		$[rhotic]_o(p^*(x))$
		⊥
		⊤
		⊥
		⊥
		$[round](x)$
		⊤
		⊥
		⊤
		$[front](x)$
		⊤
		⊥
		⊤
		⊥

$[low]_o(x)$	 [low](x) <= 0.4
	entropy = 0.4
	samples = 63
	value = [58, 5]
	True
	False
	entropy = 0.0
	samples = 58
	value = [58, 0]
	entropy = 0.0
	samples = 5
	value = [0, 5]
		

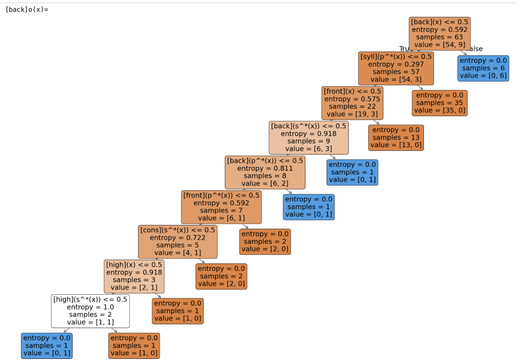
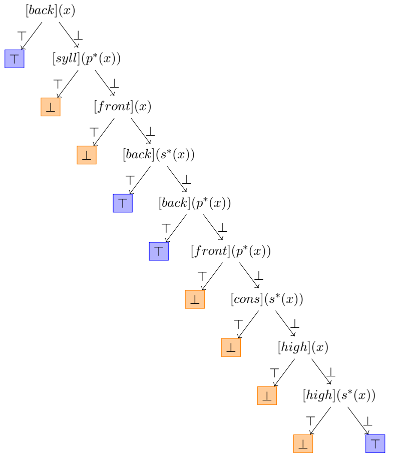
$[front]_o(x)$	 [front]o(x)=
	[front](x) <= 0.5
	entropy = 0.792
	samples = 63
	value = [48, 15]
	True
	False
	[syll](p^*(x)) <= 0.5
	entropy = 0.25
	samples = 48
	value = [46, 2]
	[syll](p^*(x)) <= 0.5
	entropy = 0.567
	samples = 15
	value = [2, 13]
	[front](p^*(x)) <= 0.5
	entropy = 0.619
	samples = 13
	value = [11, 2]
	entropy = 0.0
	samples = 35
	value = [35, 0]
	entropy = 0.0
	samples = 13
	value = [0, 13]
	entropy = 0.0
	samples = 2
	value = [2, 0]
	entropy = 0.0
	samples = 11
	value = [11, 0]
	entropy = 0.0
	samples = 2
	value = [0, 2]
		 $[front](x)$
		⊤
		⊥
		$[syll](p^*(x))$
		$[syll](p^*(x))$
		⊤
		⊥
		⊤
		⊥
		⊥
		⊤
		⊥
		$[front](p^*(x))$
		⊤
		⊥
		⊤
		⊥

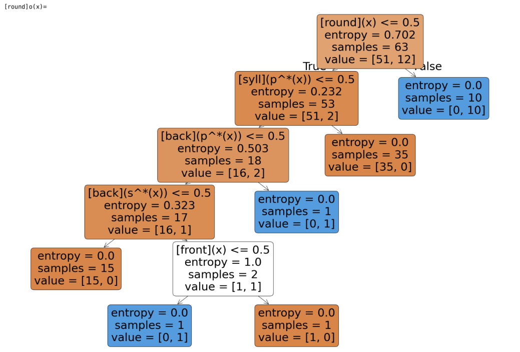
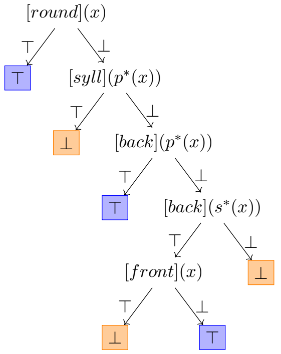
Continued on the next page

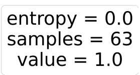
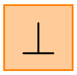
Therefore, $[cons]_o(x)$, $[COR]_o(x)$ and $[DOR]_o(x)$ always return ⊥ – this is a side effect of only including the rhymes in the dataset. [+cons] segments will still surface in the onset.

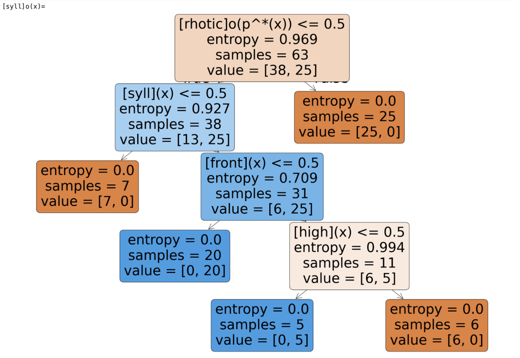
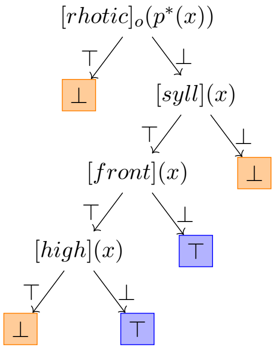
Table 9, Continued

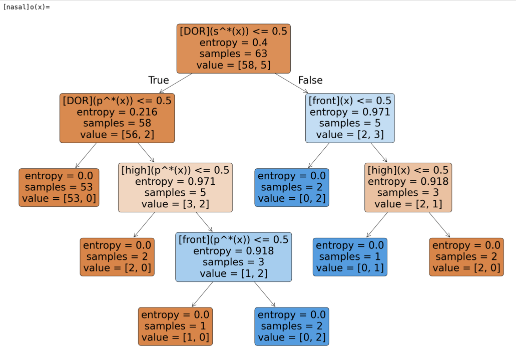
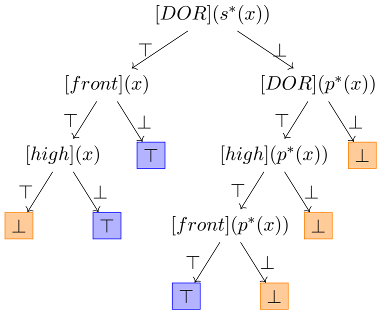
BMRS-Tree	Original	Reproduced

$[back]_o(x)$	 [back]o(x)=
	[back](x) <= 0.5
	entropy = 0.592
	samples = 63
	value = [54, 9]
	True
	False
	[syll](p^*(x)) <= 0.5
	entropy = 0.297
	samples = 57
	value = [54, 3]
	entropy = 0.0
	samples = 6
	value = [0, 6]
	[front](x) <= 0.5
	entropy = 0.575
	samples = 22
	value = [19, 3]
	entropy = 0.0
	samples = 35
	value = [35, 0]
	[back](s^*(x)) <= 0.5
	entropy = 0.918
	samples = 9
	value = [6, 3]
	entropy = 0.0
	samples = 13
	value = [13, 0]
	[back](p^*(x)) <= 0.5
	entropy = 0.811
	samples = 8
	value = [6, 2]
	entropy = 0.0
	samples = 1
	value = [0, 1]
	[front](p^*(x)) <= 0.5
	entropy = 0.592
	samples = 7
	value = [6, 1]
	entropy = 0.0
	samples = 1
	value = [0, 1]
	[cons](s^*(x)) <= 0.5
	entropy = 0.722
	samples = 5
	value = [4, 1]
	entropy = 0.0
	samples = 2
	value = [2, 0]
	[high](x) <= 0.5
	entropy = 0.918
	samples = 3
	value = [2, 1]
	entropy = 0.0
	samples = 2
	value = [2, 0]
	[high](s^*(x)) <= 0.5
	entropy = 1.0
	samples = 2
	value = [1, 1]
	entropy = 0.0
	samples = 1
	value = [1, 0]
	entropy = 0.0
	samples = 1
	value = [0, 1]
	entropy = 0.0
	samples = 1
	value = [1, 0]
		 $[back](x)$
		⊤
		⊥
		⊤
		$[syll](p^*(x))$
		⊤
		⊥
		⊥
		$[front](x)$
		⊤
		⊥
		⊥
		$[back](s^*(x))$
		⊤
		⊥
		⊤
		$[back](p^*(x))$
		⊤
		⊥
		⊤
		$[front](p^*(x))$
		⊤
		⊥
		⊥
		$[cons](s^*(x))$
		⊤
		⊥
		⊥
		$[high](x)$
		⊤
		⊥
		⊥
		$[high](s^*(x))$
		⊤
		⊥
		⊥
		⊤

$[round]_o(x)$	 [round]o(x)=
	[round](x) <= 0.5
	entropy = 0.702
	samples = 63
	value = [51, 12]
	True
	False
	[syll](p^*(x)) <= 0.5
	entropy = 0.232
	samples = 53
	value = [51, 2]
	entropy = 0.0
	samples = 10
	value = [0, 10]
	[back](p^*(x)) <= 0.5
	entropy = 0.503
	samples = 18
	value = [16, 2]
	entropy = 0.0
	samples = 35
	value = [35, 0]
	[back](s^*(x)) <= 0.5
	entropy = 0.323
	samples = 17
	value = [16, 1]
	entropy = 0.0
	samples = 1
	value = [0, 1]
	entropy = 0.0
	samples = 15
	value = [15, 0]
	[front](x) <= 0.5
	entropy = 1.0
	samples = 2
	value = [1, 1]
	entropy = 0.0
	samples = 1
	value = [0, 1]
	entropy = 0.0
	samples = 1
	value = [1, 0]
		 $[round](x)$
		⊤
		⊥
		⊤
		$[syll](p^*(x))$
		⊤
		⊥
		⊥
		$[back](p^*(x))$
		⊤
		⊥
		⊤
		$[back](s^*(x))$
		⊤
		⊥
		$[front](x)$
		⊥
		⊤
		⊥
		⊥
		⊤

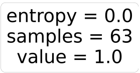
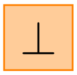
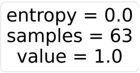
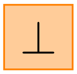
$[cons]_o(x)$		

$[syll]_o(x)$	 [syll]o(x)=
	[rhotic]o(p^*(x)) <= 0.5
	entropy = 0.969
	samples = 63
	value = [38, 25]
	[syll](x) <= 0.5
	entropy = 0.927
	samples = 38
	value = [13, 25]
	entropy = 0.0
	samples = 25
	value = [25, 0]
	entropy = 0.0
	samples = 7
	value = [7, 0]
	[front](x) <= 0.5
	entropy = 0.709
	samples = 31
	value = [6, 25]
	entropy = 0.0
	samples = 20
	value = [0, 20]
	[high](x) <= 0.5
	entropy = 0.994
	samples = 11
	value = [6, 5]
	entropy = 0.0
	samples = 5
	value = [0, 5]
	entropy = 0.0
	samples = 6
	value = [6, 0]
		 $[rhotic]_o(p^*(x))$
		⊤
		⊥
		⊥
		$[syll](x)$
		⊤
		⊥
		$[front](x)$
		⊥
		⊤
		⊥
		$[high](x)$
		⊤
		⊤
		⊥
		⊥
		⊤

$[nasal]_o(x)$	 [nasal]o(x)=
	[DOR](s^*(x)) <= 0.5
	entropy = 0.4
	samples = 63
	value = [58, 5]
	True
	False
	[DOR](p^*(x)) <= 0.5
	entropy = 0.216
	samples = 58
	value = [56, 2]
	[front](x) <= 0.5
	entropy = 0.971
	samples = 5
	value = [2, 3]
	entropy = 0.0
	samples = 53
	value = [53, 0]
	[high](p^*(x)) <= 0.5
	entropy = 0.971
	samples = 5
	value = [3, 2]
	entropy = 0.0
	samples = 2
	value = [0, 2]
	[high](x) <= 0.5
	entropy = 0.918
	samples = 3
	value = [2, 1]
	entropy = 0.0
	samples = 2
	value = [2, 0]
	[front](p^*(x)) <= 0.5
	entropy = 0.918
	samples = 3
	value = [1, 2]
	entropy = 0.0
	samples = 1
	value = [0, 1]
	entropy = 0.0
	samples = 2
	value = [2, 0]
	entropy = 0.0
	samples = 1
	value = [1, 0]
	entropy = 0.0
	samples = 2
	value = [0, 2]
		 $[DOR](s^*(x))$
		⊤
		⊥
		$[front](x)$
		$[DOR](p^*(x))$
		⊤
		⊥
		⊤
		⊥
		$[high](x)$
		⊤
		$[high](p^*(x))$
		⊥
		⊤
		⊥
		⊤
		⊥
		⊥
		⊤
		$[front](p^*(x))$
		⊥
		⊤
		⊥
		⊤
		⊥

Continued on the next page

Table 9, Continued

BMRS-Tree	**Original**	**Reproduced**
$[COR]_o(x)$		
$[DOR]_o(x)$		

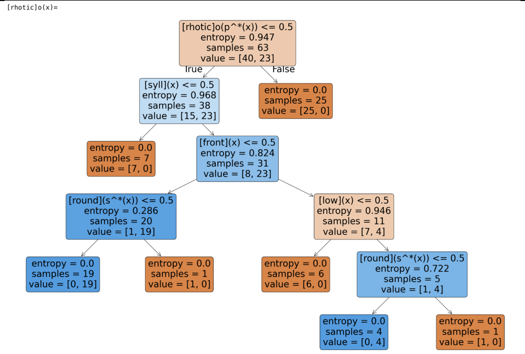
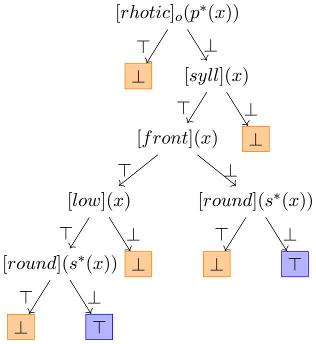
$[rhotic]_o(x)$	 [rhotic]o(x)=
	[rhotic]o(p^*(x)) <= 0.5
	entropy = 0.947
	samples = 63
	value = [40, 23]
	True
	False
	[syll](x) <= 0.5
	entropy = 0.968
	samples = 38
	value = [15, 23]
	entropy = 0.0
	samples = 25
	value = [25, 0]
	entropy = 0.0
	samples = 7
	value = [7, 0]
	[front](x) <= 0.5
	entropy = 0.824
	samples = 31
	value = [8, 23]
	[round](s^*(x)) <= 0.5
	entropy = 0.286
	samples = 20
	value = [1, 19]
	[low](x) <= 0.5
	entropy = 0.946
	samples = 11
	value = [7, 4]
	entropy = 0.0
	samples = 19
	value = [0, 19]
	entropy = 0.0
	samples = 1
	value = [1, 0]
	entropy = 0.0
	samples = 6
	value = [6, 0]
	[round](s^*(x)) <= 0.5
	entropy = 0.722
	samples = 5
	value = [1, 4]
	entropy = 0.0
	samples = 4
	value = [0, 4]
	entropy = 0.0
	samples = 1
	value = [1, 0]
		 [rhotic]o(p*(x))
		⊤
		⊥
		⊥
		[syll](x)
		⊤
		⊥
		[front](x)
		⊥
		⊤
		⊥
		[low](x)
		[round](s*(x))
		⊤
		⊥
		⊤
		⊥
		[round](s*(x))
		⊥
		⊥
		⊤
		⊤
		⊥
		⊥
		⊤

Table 9: BMRS-Tree Diagrams in Mandarin Rhotacization

The Logic of Linearization: Interpretations of Trees via Strings

Vincent Czarnecki
Rutgers University
vincent.czarnecki@rutgers.edu

Abstract

This paper introduces a novel method of linearization, casting it as a model-theoretic *interpretation*. Within Model Theory, an interpretation is a way of understanding a structure through the lens of another structure– in this sense, linearization is an interpretation of a tree's string yield through the lens of the tree. Such a formal characterization allows us to explicitly codify locality into the post-syntax (in line with Embick and Noyer (1999)). This has strong potential implications for the nature of syntax-phonology interaction in terms of formal complexity and typological predictions of phrasal phonology. Crucially, casting linearization in this way also opens the door for a closer unification of how we understand the computational properties of interfaces between linguistic modules more generally.

1 Introduction

Model Theory is a subfield within mathematical logic that is used to formally reason about structures and the properties they satisfy. There has been a rich tradition of using Model Theory within generative semantics. More recently however, research in theoretical computational linguistics has shown that Model Theory is an extremely useful tool for understanding syntax, phonology, morphology, and phonetics as well. Due to Model Theory's abstract and domain-general nature, there is a great deal of freedom in the sorts of structures that can be defined and the mappings between them, making it well-suited for linguistic theorizing.

For example, Model Theory has been used in syntax to formally reason about the computational properties of Government and Binding Theory (Rogers and Nordlinger, 1998). More recently, Model Theory has been used extensively by phonologists to understand both phonological well-formedness of structures (Strother-Garcia et al., 2016; Jardine, 2017) as well as mappings between underlying structures and surface structures (Oakden, 2021; Bhaskar et al., 2020). In Nelson (2024), model-theoretic interpretations are used to model autosegmental coupling graphs, as well as transformations between them and string representations, showing a use case in the phonetics-phonology interface. In (Petrovic, 2023), Model Theory is used to reason about the computational nature of morphological processes. In terms of complexity, this type of formalization also allows for a richer understanding of the tight relationship between learnability and computational simplicity with respect to typological predictions (Lambert et al., 2021; Rawski, 2021).

Knowing that model-theoretic representations have given novel insights to our formal understanding of separate linguistic modules, a natural question arises: *How can we use knowledge of these modules independently to understand their interaction?* Namely, if model-theoretic representations allow us to understand the formal properties of semantics, syntax, phonology, morphology, phonetics in isolation, and we know that it is extremely well-suited for understanding the relationships between different structures, then it should also serve as an invaluable tool for understanding the formal properties of their interfaces. This paper is a step in this direction, showing that linearization can be understood as an interpretation of linear post-syntactic representations through the lens of hierarchical syntactic representations. While this is one particular use case for the much broader endeavor of using Model Theory investigations of the interfaces, this opens up the door for a great body of research while making novel observations about the nature of linearization.

The paper is organized as follows. Section 2 gives an introduction to Model Theory, discussing string models and interpretations. In Section 3, we

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 123-132.
Eugene, Oregon, July 18-20, 2025

discuss linearization and show how it can be formulated as an interpretation from trees to strings and sketches an approach toward incorporating a simple case of movement into the analysis. Section 4 discusses some broader theoretical implications for this view of linearization.

2 Model Theory

A *signature* $\mathcal{S}$ is simply a collection of functions, relations and constants. The discussion here will be limited to dealing with relations, so we will stick to signatures that contain only relations, not functions or constants. A *relational model* is a pair $\langle D \mid r_1, \ldots r_n \rangle$ where D is some domain, and each r_i is a k-ary relation from the signature $\mathcal{S}$ over elements in the domain D. In place of *model*, the word *structure* is also commonly used. Here, k-ary simply means that the relation r_i takes k elements of D as its arguments. For example, $p(x)$ where $x \in D$ would be a unary relation, $q(x, y)$ where $x, y \in D$ would be a binary relation, etc. The focus of this section is on using these models to define strings and mappings between them.

2.1 Strings

Consider the string *apba*. It contains only the segments {*a,b,p*} and there are four elements, the first bearing *a*, the second bearing *p*, the third bearing *b*, and the fourth bearing *a*. Let the domain $D = \{0, 1, 2, 3\}$ represent the indices of the string and the alphabet (set of symbols) $\Sigma = \{a, b, p\}$ represent the labels each index can bear. For each of these symbols, define a unary relation that indicates whether or not an index x of the string bears that symbol: so there are three unary relations $a(x), b(x), p(x)$. For our string *apba*, it is the case that $a(0)$, $p(1)$, $b(2)$, and $a(3)$ are all true, and any of these relations for other domain elements will be false. Each index bears a label, but there must be some way to tell what precedes what in our string. This can be done by relating the indices through a binary precedence relation. Strict precedence $\lhd(x, y)$ states that x comes before y with nothing else in between. A string model for *apba* using strict precedence $\lhd(x, y)$ is shown below:

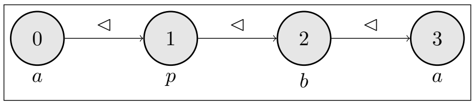


Figure 1: String Model with Strict Precedence

Alternatively, one could use general precedence $< (x, y)$ where x comes before y at any point in the string. Whether a structure is defined using general precedence or strict precedence directly affects the sorts of generalizations one can make. For example, using strict precedence it is natural to ban immediately adjacent segments like a ban on any obstruent immediately following a nasal (say, a *NT constraint), whereas using general precedence it is natural to ban sequences of segments like long distance sibilant harmony that bans an ʃ following an s anywhere in the string (say, a *s . . . ʃ constraint). For a broader picture of how representations and the nature of constraints relate to one another in phonology, see Heinz (2018). This difference will have important implications for the motivation of the view of linearization argued for here.

2.2 Interpretations

Informally, a logical interpretation is a mapping that takes an input structure Σ in a signature $\mathcal{S}$ and uses logical expressions to recast it as an output structure Γ in a signature $\mathcal{G}$, shown abstractly in Figure 2. One way to imagine this is interpreting an output structure Γ *through the lens of an input structure* Σ. It is also convenient to imagine this as a transformation, where an input structure is transformed into an output structure. Here, the term *logical transduction* is used.

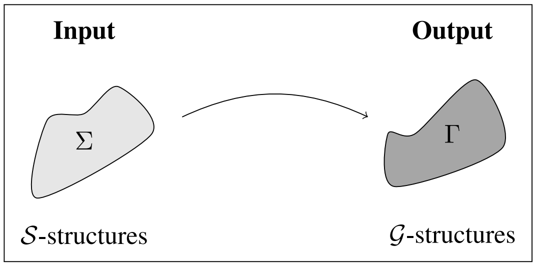


Figure 2: General Sketch of a Logical Transduction

Let $\mathcal{G}$ be our output signature with relations $r_1^{'}, \ldots, r_n^{'}$. For each relation $r_i^{'}$ in the output signature $\mathcal{G}$, there must be a definition which is defined using only relations r_i from the input signature $\mathcal{S}$ or more complex helper predicates constructed from them. There is also a copyset $C = \{1, \ldots, m\}$ that copies pieces of the input structure to be (potentially) used by the output structure. Essentially for each node x in the input structure Σ, depending on the size of the copyset, a corresponding copy is used: there can be an x^0 copy, x^1

copy, x^2 copy, and so on up to m, meaning that the input structure will grow linearly depending on the size of the copyset.

Consider the input signature $\mathcal{S}$ (from the string model for *abpa* in Figure 1) and the output signature $\mathcal{G}$, containing precedence relations that mediate precedence between copies:

$$\mathcal{S} = \{a(x), b(x), p(x), \lhd(x,y)\}$$
$$\mathcal{G} = \{a^0(x), p^0(x), b^0(x), a^1(x), p^1(x), b^1(x), \lhd^{0,0}(x,y), \lhd^{0,1}(x,y), \lhd^{1,0}(x,y), \lhd^{1,1}(x,y)\}$$

In the output signature $\mathcal{G}$, the relations $\sigma^0(x)$ mean that x's 0-th copy is labeled with the symbol σ and $\sigma^1(x)$ mean that x's 1-st copy is labeled with the symbol σ. The relations $\lhd^{i,j}(x,y)$ mediate strict precedence *between different copies* in the output. In other words, $\lhd^{i,j}(x,y)$ means that x's i-th copy strictly precedes y's j-th copy in the output. This is made clear in the example that follows.

Using these two signatures, we will construct an input $\mathcal{S}$-structure Σ, an output $\mathcal{G}$-structure Γ, and an interpretation will be constructed between them that epenthesizes a's between a p followed by a b. This can be written as a standard rewrite rule $\varnothing \rightarrow a/p\_b$. Note that a copyset of $C = \{0, 1\}$ is needed because the string will grow in length by one node any time there is a '*pb*' substring. We proceed by defining each relation in the output signature using relations from the input signature. The interpretation is shown pictorially in Figure 3.

For the labeling relations, every node in the 0-th copy is going to remain faithful to the input. Nothing is deleted, there are only things to add and so this copy remains the same. In the 1-st copy, only nodes labelled with an a will ever appear since that is the only segment we wish to add (since b's or p's will never be epenthesized). For the precedence relations, strict precedence will hold between two nodes in the 0-th copy if they aren't a p strictly followed by a b. The only time a 0-th copy will strictly precede a 1-st copy is when there is an a being inserted, namely between a p strictly followed by a b. The only time a 1-st copy will precede a 0-th copy is when it is the b in the configuration just described. There will never be strict precedence between elements both in the 1-st copy, since this would correspond to adding two a's in a row.

In Figure 3, the dashed nodes represent copies of nodes that are not used in the interpretation. When the copyset is constructed, all copies *have the potential to be used*, but the actual definitions of the labeling and precedence relations determine which are actually used. In this mapping, an input string $apba$ will map to $apaba$, since the a was epenthesized between the p and b, whereas an input string $abapa$ would simply map to $abapa$ since there are no 'pb' substrings.

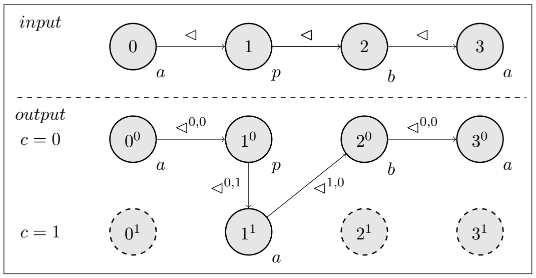


Figure 3: a-epenthesis between p and b

Thinking more generally, this is an example of how an output string has a particular form *based on specified conditions on its corresponding input string*. Thus, we are interpreting the output string through the lens of the input string. Shifting focus to linearization, an output string structure has a particular form based on specified conditions on its corresponding input tree structure. To understand this more clearly, tree models must first be defined.

3 Linearization as an Interpretation

3.1 Tree Models

It is standard practice within model-theoretic syntax to define trees with respect to a domain $D \subsetneq \mathbb{N}$ of nodes, a binary general dominance relation $\lhd^*(x,y)$ and a left-of/precedence relation $\prec (x,y)$ as in Rogers and Nordlinger (1998). There are many theoretical reasons to suggest that they should instead be defined over something more closely resembling syntactic selection instead of a precedence relation, but in order to keep the discussion more tractable, this convention serves as a suitable starting point. Some ways that this can be embellished for a more well-rounded account will be discussed in later sections.

Note that the domain ranges over the natural numbers $\mathbb{N}$, but the order that they appear doesn't matter so long as the relations are consistently defined. For convenience, the convention here reflects the order that they are introduced to the derivation, assuming a bottom up derivation.[1]

[1] One could also choose to use Gorn addresses as in Lam-

There must also be labels for the nodes of our tree, so let Σ_{syn} be an input alphabet of labels for nodes of our tree. Since our nodes can bear a wide range of different syntactic properties, this alphabet can be partitioned into the following sets of lexical labels, categories, features, and movement-licensing features:

- $L = \{\text{THE}, \text{MAN}, \text{LOVES}, \text{CAKE}, \ldots\}$
- $C = \{\text{v}, \text{V}, \text{C}, \text{D}, \text{N}, \text{PERF}, \ldots\}$
- $F = \{\text{SG}, \text{PL}, 1, \ldots\}$
- $LIC = \{\texttt{+wh}, \texttt{-wh}, \texttt{+nom}, \texttt{-nom}, \ldots\}$

Thus, $\Sigma_{syn} = L \cup C \cup F \cup LIC$, and each $\sigma \in \Sigma_{syn}$ has a corresponding unary relation $\sigma(x)$ specifying some piece of syntactic information. This is one particular choice of how to encode this information relationally, inspired by Minimalist Grammars (Stabler, 1996), but many other options are available. While this is not strictly necessarily, a labelless syntax is assumed (Collins, 2002), such that non-terminal nodes without *lexical* labels (bearing no $\sigma \in L$) represent instantiations of Merge.[2]

We will start with a simplified, abstract example for clarity and it will be expanded when movement is discussed. Consider the input signature:

$$\Sigma = \{\lhd^*(x,y), \prec(x,y), \sigma_i(x)\}$$

where:

- $\lhd^*(x,y)$ is the binary general dominance relation
- $\prec(x,y)$ is the binary, asymmetric precedence relation
- $\sigma_i(x)$ are unary relations for every $\sigma_i \in \Sigma_{syn}$

To keep the discussion tractable while introducing the main properties of linearization, only lexical labels are encoded in this structure, but the general points about how labels carry over to the output structure hold for the other category and feature labels. A simplified example of an $\mathcal{S}$-structure in the signature Σ is shown in Figure 4 over the arbitrary, abstract alphabet $\Sigma_{syn} = \{\text{THE}, \text{MAN}, \text{LOVES}, \text{THE}, \text{CAKE}\}$.

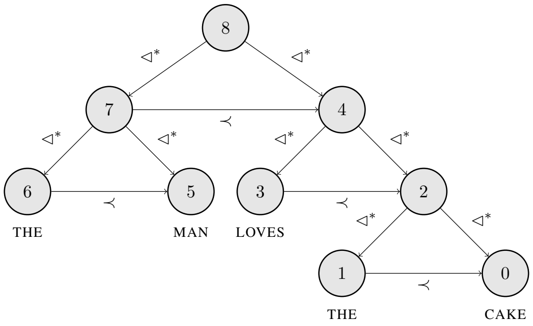


Figure 4: Linearization Toy Example

Considering this example, setting aside the issue of movement for later, the linearization intuitively yields the string "THE MAN LOVES THE CAKE". However, this is a nontrivial task since branches can be of arbitrary finite length. The next section lays out an interpretation using First-Order Logic to yield a string defined using strict precedence.

3.2 Remarks on Linearization

The main contribution of this work is to show that linearization can be concisely understood as an interpretation between trees and strings. In order to formalize this, it is crucial to first establish some theoretical assumptions of both input trees and output strings.

There is a rich body of work debating the status of linearity and recursion and their presence in syntax and phonology (Scheer, 2012, 2023; Idsardi and Raimy, 2013; Idsardi, 2018; Elfner, 2015; Ito and Mester, 2012; Cheng and Downing, 2021; Miller and Sande, 2021). This paper adheres to the view that (i) *narrow syntax contains recursion but lacks linearity* and (ii) *phonology contains linearity but lacks recursion*. To understand this, looking at work by Idsardi and Raimy (2013) is helpful. They outline three types of linearization, one of which, *immobilization*, plays a key role here. Immobilization transforms hierarchical structures built via Merge into ordered structures by introducing adjacency relations. There is a subtle but crucial point here with respect to the status of linearity in the computation of narrow syntax. The structure building taking place during narrow syntactic computation is blind to linearity, but linearity is a necessary reflex of externalization given

bert et al. (2021), where domain elements are strings in $\{0,1\}^*$ where a 0 indicates a left child and a 1 indicates a right child.

[2] A series of well-formedness conditions can be defined that more accurately reflect standard syntactic assumptions (the nodes that select project, encoding feature percolation, etc.), some of which will be explored later with respect to movement.

the temporal nature of the speech stream. So there must be a stage after syntactic structures are built which imposes linearity, and this is precisely the function of immobilization. The finer details of immobilization are beyond the scope of this paper, but similar model-theoretic tools are well-suited to formalize it. In this framework, we assume that "flattening" occurs after adjacency relations are established. Thus, the input trees of our linearization are the recursive hierarchical trees built by the narrow syntax *once they have been embellished with adjacency information*, hence the use of the precedence relation $\prec(x, y)$.

In what follows, we define this mapping using First-Order Logic, ensuring that the process remains sufficiently restrictive from a computational perspective. This formulation allows linearization to be expressed in a purely declarative manner rather than as a derivational process. It also fundamentally codifies the notion of locality into the representation, which is known to be important for the post-syntax (Embick and Noyer, 1999).

3.3 Tree-Flattening as an Interpretation

Consider an input $\mathcal{S}$-structure, a tree denoted Σ, and an output string $\mathcal{G}$-structure, a string denoted Γ, representing the concatenation of Σ's leaves in the correct order. Recall that the relations of our output string must be defined in terms of those input relations (namely, $\lhd^*, \prec, \sigma_i$ or helper predicates built using these) and this is precisely the sense in which the output string is being interpreted in terms of the input tree.

As before, two pieces are necessary: (i) which nodes from the input are relevant for the output and (ii) how they are ordered with respect to each other. The ordering will be a relation called $\texttt{lin}(x, y)$ to indicate that x and y in the input tree meet the conditions for x to strictly precede y in the linearized output string. Intuitively, only the leaves will be contained in the output structure, but the ordering between them may not be readily clear at first glance. Taking the tree in Figure 4, its intended linearization shown pictorially below in Figure 5. The example will proceed by reasoning why the ordering is the way it is, which will lead to the formal definition.

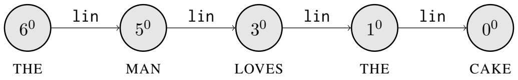


Figure 5: Output of Linearization Toy Example

We only want to include leaf nodes in our output string, and because the input will not grow in the output, we only need a single copy set $C = \{0\}$. In fact, this interpretation can be seen as a mapping that "forgets" the hierarchical information and "connects" the leaves in the correct order via linear precedence. We define a predicate $\texttt{leaf}(x) := \neg\exists y[\lhd^*(x, y)]$ that says a node x is a leaf node iff there is no node y that it dominates. Thus, the labeling relations will take the following form for each item in the input alphabet $\sigma_i \in \Sigma_{syn}$:

$$\begin{aligned}\textsc{the}^0(x) &:= \textsc{the}(x) \wedge \texttt{leaf}(x)\\ \textsc{man}^0(x) &:= \textsc{man}(x) \wedge \texttt{leaf}(x)\\ &\vdots\end{aligned}$$

To better understand why the output string has the linear order it does, some more helper predicates are defined. A $\texttt{left-leaf}$ is a leaf that has nothing preceding it, and a $\texttt{right-leaf}$ is a leaf that precedes nothing. Formally,

$$\begin{aligned}\texttt{left-leaf}(x) &:= \texttt{leaf}(x) \wedge \neg\exists y[\prec(y, x)]\\ \texttt{right-leaf}(x) &:= \texttt{leaf}(x) \wedge \neg\exists y[\prec(x, y)]\end{aligned}$$

Using these, we can define predicates to indicate whether a given node is the *left-most leaf* of a particular node, and another to indicate if a given node is the *right-most leaf* of a particular node. For a given node, whichever node is the (unique!) leaf below it such that nothing is further left is its left-most leaf and whichever node is the (unique!) leaf below it such nothing is further right is its right-most leaf.

The relevance of these becomes clear when thinking about where $\texttt{lin}(x, y)$ holds true in the tree in Figure 4. Let's observe each case: First, $\texttt{lin}(6, 5)$ because both 6 and 5 are leaves and $\prec(6, 5)$. Next, $\texttt{lin}(5, 3)$ because there is a node whose right-most leaf is 5 and it precedes a node whose left-most leaf is 3, so no other leaves can be in between them. Next, $\texttt{lin}(3, 1)$ because 3 precedes a node whose left-most leaf is 1. Finally, $\texttt{lin}(1, 0)$ for the same reason $\texttt{lin}(6, 5)$, namely both are leaves and $\prec(1, 0)$.

Thus, in all of these scenarios, expressing strict precedence in the output requires reference to left-most and right-most leafhood. Every node has a left-most and right-most leaf, and every leaf node is its own right-most and left-most leaf (since dominance is taken to be reflexive). Defining one more helper predicates aids in readability. The following predicate indicates that a node y is dominated by x and dominates z and so we say that y is *between* x and z in the tree:

$$\texttt{between}(x, y, z) := \lhd^*(x, y) \wedge \lhd^*(y, z)$$

Now having seen the importance of these configurational relationships to linearization, the formal definitions for right-most and left-most leafhood are as follows:

- A node x is the left-most leaf of a node y iff for all the left-leaf nodes z that y dominates, the only one with nothing further left is x:

$$\begin{aligned}\texttt{lml}(x, y) := \forall z[(\lhd^*(y, z) \wedge \texttt{left-leaf}(z) \\ \wedge \forall s[\texttt{between}(y, s, z) \\ \wedge \neg\exists t[\prec(t, s)]]) \leftrightarrow z = x]\end{aligned}$$

- A node x is the right-most leaf of a node y iff for all the right-leaf nodes z that y dominates, the only one with nothing further right is x:

$$\begin{aligned}\texttt{rml}(x, y) := \forall z[(\lhd^*(y, z) \wedge \texttt{right-leaf}(z) \\ \wedge \forall s[\texttt{between}(y, s, z) \\ \wedge \neg\exists t[\prec(s, t)]]) \leftrightarrow z = x]\end{aligned}$$

Now that these have been given, note that each of the cases above made some mention of x and y being the left-most or right-most leaf of two higher nodes where one precedes the other, we can call these t and s.[3] Any of these configurations leading to x strictly preceding y in the output string can be condensed into the following single condition, also shown pictorially in Figure 6:

$$\texttt{lin}(x, y) := \exists t \exists s[\prec(t, s) \wedge \texttt{rml}(t, x) \wedge \texttt{lml}(s, y)]$$

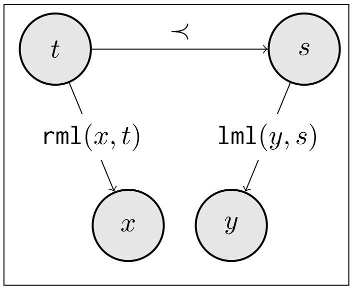


Figure 6: Conditions for Strict Precedence in Output

What we have done is reduced precedence between any two nodes in the output string to a *single declarative condition* between nodes the input tree: the node x will strictly precede y iff this condition holds. One of the primary strengths of this result is that it doesn't cast linearization in terms of a procedure, but rather it reduces it to underlying knowledge about the structural relationship between linguistic elements. Another critical property of this method of linearization is that it is definable using First-Order Logic, which is desirable from a formal complexity standpoint. This is because it limits its use of quantification to individual elements as opposed to sets of elements as would be the case in Monadic Second Order Logic. This is a nice result with respect to computational complexity, since it is subregular.

There is an important question regarding the choice of strict precedence in the output string. Recall from the earlier discussion of strings that the choice of representation (strict or general precedence) affects the available generalizations one can define. Using strict precedence it is natural to ban immediately adjacent segments like a ban on any obstruent immediately following a nasal (say, a *NT constraint). For example, with an alphabet of $\Sigma = \{\text{V}, \text{N}, \text{T}, \text{D}\}$, such a constraint would accept the string VNDV but reject the string *VNTV. In fact, this is a strictly local constraint since it depends only on a window of two elements (Chandlee, 2014). In contrast, using general precedence it is natural to ban sequences of segments like long distance sibilant harmony that bans an ʃ following an s anywhere in the string (say, a *s . . . ʃ constraint). For example, with an alphabet of $\Sigma = \{$ʃ,s,v,c$\}$, such a constraint would accept the string ʃcvcvʃ but reject the string *scvcvʃ. This is not strictly local because it contains a dependency between elements that can occur arbitrarily far away from one another. [4] The choice of strict precedence in the definition of linearization here formally hard-codes locality into our post-syntactic representations. The output of our linearization is a string of nodes labeled with morphosyntactic information and if they are related via strict precedence, this prunes out arbitrary, word-level parallels of these long distance generalizations. Thus, post-syntactic operations at

[3] Since any leaf is its own left-most leaf and right-most, it can be true that either $t = x$ or $s = y$ or both.

[4] However, it is possible to define Tier-Based Input or Output Strictly Local functions that have a relativized form of adjacency via a particular feature or category, thus naturally constraining the ability to make long-distance generalizations.

this level of representation can be modeled using ISL functions.

This simplified example did not contain any feautural information, but this will become relevant when sketching a potential analysis implementing movement. As a start, encoding standard syntactic mechanisms like selectional requirements and feature percolation can be stated as well-formedness conditions on our input trees. As an example, suppose we had a well-formedness condition in our trees that said a non-terminal node only bears a category label, for example D, iff it has two children x, y where x shares the category D and $\prec (x, y)$, which enforces that the selecting node will project its features to its parent. Another example, suppose we define a well-formedness condition for movement features `f` which states that if a leaf node bears a `-f` feature, this `-f` feature must percolate upward to its maximal projection. These are some ways to understand how this method of linearization could be expanded going forward for a more all-encompassing account.

3.4 Incorporating Movement

There are many ways one could imagine incorporating movement to this analysis. One potential way is to assume that we have a tree-to-tree mapping, where the input tree is a pre-movement tree and the output is a post-movement tree. While this does split the division of labor, a notable drawback of this approach is that it would require two separate interpretations: one solely for completing movement and another for linearization. There is also the question of how to encode movers. This could be done by embellishing the alphabet with movement traces, where our trees would instead have trace labels at the launching sites and lexical labels at their landing sites. This would drastically increase the length of the alphabet since this would presumably require a trace corresponding to each label already in L.

Another potential alternative would entail altering some of our representational assumptions for input trees. Our input trees could be modified to include a separate relation to encode movement. For example, suppose we had a relation $\mathcal{M}(x, y)$ where x is the the highest node of a mover and y is a node immediately dominating a movement attracting head. This could then be used to define a structural input condition to determine the placement of x's children in the output string.

The alternative sketched here assumes movement takes place concurrently with linearization, sketched using an example in Figure 7. While an analysis in which trees are built with a syntactic selection relation as opposed to precedence may reflect the nature of syntactic computation more accurately, this would be beyond the scope of this paper. Incorporating the exhaustive well-formedness conditions, movement configurations, successive cyclic movement or enforcing relativized minimality in all generality would be considerably much more involved than is possible here, but such an analysis is left for further work. Given the fact that most work in model-theoretic syntax has assumed the sorts of representations used here, this is sufficient for the central points regarding linearization.

In the tree in Figure 7, substructures that consist of movers are darkened for clarity. In the output string, the string yield of the movers is outlined with a dashed box to clarify that these are the leaves of an *entire substructure* with relevant properties from the input. There are two moving substructures in this tree. One is the substructure with the root 9, the phrase "THE MAN", driven by a `nom` feature and the other is the substructure with the root 2, the phrase "WHICH CAKE", driven by a `wh` feature. In the output string, the moved phrase driven by `nom` appears to the left of the attracting head T bearing a `+nom` feature. Similarly, the moved phrase driven by `wh` appears to the left of the attracting head C bearing a `+wh` feature bearing a `-wh` feature.

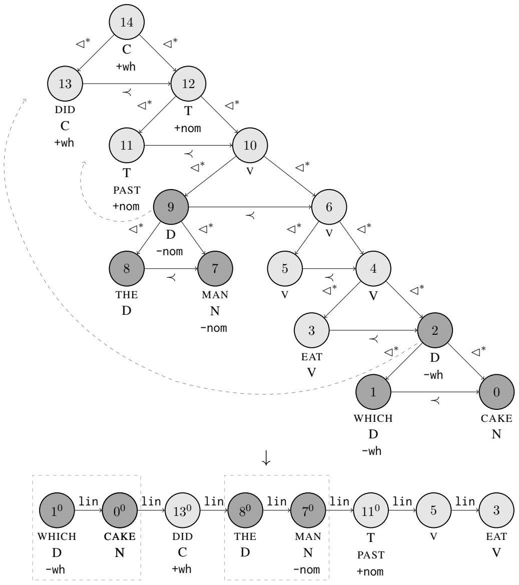


Figure 7: Linearization Toy Example with Movement

What is true about each of these moving substructures with respect to these heads? Each of them are rooted with a node that bears a `-f` feature for some movement-driving feature `f`, as per the well-formedness condition posited earlier. Thus, the yield of this substructure should occur before the `f`-movement driving head in the output string, meaning precisely that the `rml` of the moving substructure will strictly precede this `+f` head. In the case of the `nom` movement, the output string structure will have 7 (the MAN-bearing node) strictly preceding 11 (the `+nom`-bearing T node). Similarly, in the case of the `wh` movement, the output string structure will have 0 (the CAKE-bearing node) strictly preceding 13 (the `+wh`-bearing C node). Together with the `lin`(x, y) condition, this covers the movers themselves and their relationship to the movement-driving heads.

There are two remaining tasks: we must determine what precedes the mover once it lands and the nodes around its launching site. Firstly, it must be ensured that the mover's `lml` comes before the unique node which would have met `lin`(x, y) in the input (where y is the movement attracting head). For example, C will strictly precede the `nom`-mover's `lml` bearing THE. Similarly, the node bearing WHICH will be the first node in the string since there is nothing higher than the attracting head. Secondly, it must be ensured that the nodes surrounding the mover, if they exist, are connected via strict precedence. In other words, the next highest node that would have met `lin`(x, y) where y is the `lml` of the mover should strictly precede the next lowest node that would have met `lin`(x, y) where x is the `rml` of the mover. For example, the node bearing T will strictly precede the node bearing V since they "surround" the launching site. Similarly, the node bearing EAT will be the final node in the string since there is nothing lower than the mover in the input tree.

This is only sketched out as an example, but the entirety of the mapping just described is definable by making modifications to the `lin`(x, y) condition within First-Order Logic. This is because in what was just described, it only requires quantification of individual nodes, not arbitrary subsets of nodes, leaving the definition within First-Order Logic. Even though this substructure can be arbitrarily large, the only relevant nodes of the mover to be picked out are its root, `rml` and `lml` and nodes in between are covered by `lin`(x, y). An anonymous reviewer points out that the scope here is limited to a relatively simple case of movement, but further work could provide a more structured analysis of more complex cases (e.g. multiple movers attracting to a single head, smuggling, remnant movement, mixed-headedness, etc.) using these tools and examine whether they remain within First-Order Logic.

4 Discussion

This novel view of linearization comes with many theoretical advantages. Firstly, it was shown (albeit through automata-theoretic as opposed to model-theoretic means) that Recursive Prosody is non finite state and thus requires more computational power (Dolatian et al., 2021). This declarative *linearization-as-flattening* approach has the benefit that it only uses First-Order Logic, which is notably less computationally expensive than Monadic Second-Order Logic, which is required for mappings that are finite state or more powerful.

This approach may have interesting implications for our view of the syntax-phonology interface regarding the status of recursion in phonology. For a recent view on the debate of the status of recursive prosodic approaches and procedural approaches, see Lee and Selkirk (2022); Newell and Sailor (in press). It is well-known that there are often mismatches between syntactic and phonological domains (Cheng and Downing, 2016); however, these mismatches often appear to occur at or very near to Spell-out boundaries. Accommodating this notion of "at or very near to" is extremely amenable to this type of analysis given its inherent locality properties. If string yields are embellished with boundary information (either by means of boundary symbol like $\rtimes$ or $\ltimes$, respectively or relations that hold of a node $\varphi_{\texttt{init}}(x)$ or $\varphi_{\texttt{fin}}(x)$, respectively), then it may be expected that the range of syntax-phonology mismatches are accounted for through by employing Input Strictly Local (ISL) restructuring functions (Chandlee, 2014), which are very computationally restrictive. Dobashi (2003, 2019) has work detailing phonological domain restructuring and its typological implications. These approaches are nicely compatible and would serve as a fruitful integration of theoretical and computational results at the syntax-phonology interface, creating a new avenue for more formal analyses in this domain.

There are other computational characterizations of linearization that exist currently; for example, (Graf, 2022a,b) gives an elegant formal characterization which is ISL, a strikingly desirable property with respect to computational complexity; however, this does require abiding by quite strong representational assumptions about the nature of trees that are undoubtedly formally well-founded and rigorous, but have not received a wider adoption in more general syntactic literature. The analysis makes use of dependency trees, which are relatively uncommon outside of the space of computational syntax. While Graf defines a straightforward mapping between more standard phrase structure trees and dependency trees, the analysis proposed here takes a view where linearization occurs straight from more standard syntactic representations dispensing with the need for such intermediate mappings. This also adds to a recent body of work that has begun to bridge the gap between theoretical work on the interface and separate computational work in phonology and syntax (Dolatian et al., 2021; Yu, 2021; Vu et al., 2022; Stabler and Yu, 2023), despite some of the differing theoretical assumptions regarding the status of recursion.

5 Conclusion

This paper has presented a novel method of linearization, casting it as a model-theoretic interpretation between strings and trees. It is both computationally restrictive and hard-codes locality into the output string representations, all while expressing ordering between nodes as a single declarative condition. A potential expansion incorporating movement was explored through a motivating example, showing that the tools are amenable to further modifications. The most central advantage to this analysis is the fact that it is a step toward computationally unifying how we think about linguistic modules and their interaction, despite some of their representational differences.

Acknowledgements

I would like to thank Adam Jardine and the attendees of the 2024 Rutgers Subregular Phonology Workshop for their valuable insights and discussion on this work. I would also like to thank the three anonymous reviewers for their thoughtful and constructive feedback.

References

Siddharth Bhaskar, Jane Chandlee, Adam Jardine, and Christopher Oakden. 2020. Boolean monadic recursive schemes as a logical characterization of the subsequential functions. In *Language and Automata Theory and Applications: 14th International Conference, LATA 2020, Milan, Italy, March 4-6, 2020, Proceedings 14*, pages 157–169. Springer.

Jane Chandlee. 2014. *Strictly local phonological processes*. University of Delaware.

Lisa Lai-Shen Cheng and Laura J Downing. 2016. Phasal syntax= cyclic phonology? *Syntax*, 19(2):156–191.

Lisa Lai-Shen Cheng and Laura J Downing. 2021. Recursion and the definition of universal prosodic categories. *Languages*, 6(3):125.

Chris Collins. 2002. Eliminating labels. *Derivation and explanation in the Minimalist Program*, pages 42–64.

Yoshihito Dobashi. 2003. *Phonological phrasing and syntactic derivation*. Cornell University.

Yoshihito Dobashi. 2019. *Externalization: Phonological interpretations of syntactic objects*. Routledge.

Hossep Dolatian, Aniello De Santo, and Thomas Graf. 2021. Recursive prosody is not finite-state. In *Proceedings of the Seventeenth SIGMORPHON Workshop on Computational Research*.

Emily Elfner. 2015. Recursion in prosodic phrasing: Evidence from connemara irish. *Natural Language & Linguistic Theory*, 33:1169–1208.

David Embick and Rolf Noyer. 1999. Locality in post-syntactic operations. *MIT working papers in linguistics*, 34(265-317).

Thomas Graf. 2022a. Diving deeper into subregular syntax. *Theoretical Linguistics*, 48(3-4):245–278.

Thomas Graf. 2022b. Subregular linguistics: bridging theoretical linguistics and formal grammar. *Theoretical Linguistics*, 48(3-4):145–184.

Jeffrey Heinz. 2018. *The computational nature of phonological generalizations*, pages 126–195. De Gruyter Mouton, Berlin, Boston.

William Idsardi and Eric Raimy. 2013. Three types of linearization and the temporal aspects of speech. *Challenges to linearization*, 1:31–56.

William J. Idsardi. 2018. *Why Is Phonology Different? No Recursion*, pages 212–223. Cambridge University Press.

Junko Ito and Armin Mester. 2012. Recursive prosodic phrasing in japanese. *Prosody matters: Essays in honor of Elisabeth Selkirk*, pages 280–303.

Adam Jardine. 2017. On the logical complexity of autosegmental representations. In *Proceedings of the 15th Meeting on the Mathematics of Language*, pages 22–35.

Dakotah Lambert, Jonathan Rawski, and Jeffrey Heinz. 2021. Typology emerges from simplicity in representations and learning. *Journal of Language Modelling*, 9.

Seunghun J. Lee and Elisabeth Selkirk. 2022. Xitsonga tone: The syntax-phonology interface. In *Prosody and Prosodic Interfaces*. Oxford University Press.

Taylor L Miller and Hannah Sande. 2021. Is word-level recursion actually recursion? *Languages*, 6(2):100.

Scott Nelson. 2024. *The Computational Structure of Phonological and Phonetic Knowledge*. Ph.D. thesis, State University of New York at Stony Brook.

Heather Newell and Craig Sailor. in press. *Minimalism and the Syntax-Phonology Interface*. Cambridge University Press.

Christopher Donal Oakden. 2021. *Modeling phonological interactions using recursive schemes*. Ph.D. thesis, Rutgers The State University of New Jersey, School of Graduate Studies.

Andrija Petrovic. 2023. *Insights From the Interfaces: Morphological Processes as String Transductions*. Ph.D. thesis, State University of New York at Stony Brook.

Jonathan Rawski. 2021. *Structure and Learning in Natural Language*. Ph.D. thesis, State University of New York at Stony Brook.

James Rogers and Rachel Nordlinger. 1998. *A descriptive approach to language-theoretic complexity*. Citeseer.

Tobias Scheer. 2012. Chunk definition in phonology: prosodic constituency vs. phase structure. *Modules and interfaces*, pages 221–253.

Tobias Scheer. 2023. Recursion in phonology: Anatomy of a misunderstanding. *Representing phonological detail: Segmental structure and representations*, pages 265–287.

Edward Stabler. 1996. Derivational minimalism. In *International conference on logical aspects of computational linguistics*, pages 68–95. Springer.

Edward P Stabler and Kristine M Yu. 2023. Unbounded recursion in two dimensions, where syntax and prosody meet. *Proceedings of the Society for Computation in Linguistics*, 6(1):343–356.

Kristina Strother-Garcia, Jeffrey Heinz, and Hyun Jin Hwangbo. 2016. Using model theory for grammatical inference: a case study from phonology. In *Proceedings of The 13th International Conference on Grammatical Inference*, volume 57 of *JMLR: Workshop and Conference Proceedings*, pages 66–78.

Mai Ha Vu, Aniello De Santo, and Hossep Dolatian. 2022. Logical transductions for the typology of ditransitive prosody. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 29–38.

Kristine M Yu. 2021. Computational perspectives on phonological constituency and recursion. *Catalan journal of linguistics*, 20:77–114.

Adjunction in (T)SL Syntax

Kenneth Hanson
Department of Linguistics
Stony Brook University
mail@kennethhanson.net

Abstract

Adjunction is intuitively a local operation, yet its subregular complexity is dependent on both the geometry of the syntactic representation as well as the specific model of adjunction assumed. Here, I propose a model of adjunction which is strictly local (SL) over Minimalist Grammar (MG) dependency trees, and which incorporates the core properties of optionality, iteration, invisibility to selection, and adjunct ordering restrictions. Non-locality is avoided in cases of recursive adjunction, and an interesting treatment of several other formal properties of adjunction is made possible.

1 Introduction

In the last several years, a two-level classification of the computational complexity of syntax has emerged: local dependencies such as selection are *strictly local* (SL) over trees, while non-local dependencies such as movement, agreement, and case assignment are *tier-based strictly local* (TSL), a straightforward generalization of SL in which a subset of non-salient elements are ignored (Graf, 2018, 2022b; Hanson, 2023b, 2025; Vu et al., 2019). This closely matches past results on local and non-local phonological dependencies, which are predominantly SL and TSL over strings, respectively (Heinz, 2018), providing evidence of cognitive parallelism across linguistic domains (Graf, 2022a).

The placement of adjunction within this scheme, however, has remained unclear, as formal models of adjunction vary in their subregular complexity (Graf, 2014). Furthermore, the complexity of adjunction interacts with that of selection: in the derivation tree language for a Minimalist Grammar (MG) with recursive adjunction, the complexity of selection is increased to TSL (Graf, 2018). This is not a terrible state of affairs, as it would mean that the overall complexity of much of syntax is quite low, and uniform across operations. At the same time, selection is typically considered to be highly local. For example, a verb may select the category of its complement, but not the complement of its complement, let alone more distant items, yet this is exactly what we would predict if selection was TSL. Similarly, most of the key properties of adjunction require only a SL grammar (Hanson, 2023a). We therefore ask: can the non-locality of adjunction, and by extension selection, be eliminated?

The answer is affirmative. With minor adjustments, the *MG dependency tree model* defined in Graf and Kostyszyn (2021) can easily accommodate a linguistically satisfactory SL model of adjunction, which includes the core properties of optionality, iteration, invisibility to selection, and ordering restrictions among adjuncts. The primary change required is to generalize the model to *unranked* trees, which have no maximum branching factor. This is highly natural from a mathematical perspective, and brings several added benefits. Selection remains SL, as does the combined grammar for selection and adjunction, even allowing for a degree of variation in the position of adjunction. The model also provides an interesting perspective on the distinction between left and right adjuncts which suggests doubling down on separation between dependency structure and constituency structure, relegating the latter to the post-syntactic map.

The remainder of this paper proceeds as follows. First, I introduce the necessary background on adjunction, MG dependency trees, and strictly local string and tree languages (2). Next, I implement a strictly local grammar for MG dependency trees which includes selection as well as adjunction in the style of Frey and Gärtner (2002) (3). From there, I refine the system to incorporate recursive adjunction (4) and adjunct ordering restrictions (5), building on insights from Graf (2018) and Fowlie (2013). Finally, I address some alternatives and potential complications for the proposed model, and directions for future research (6).

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 133-143.
Eugene, Oregon, July 18-20, 2025

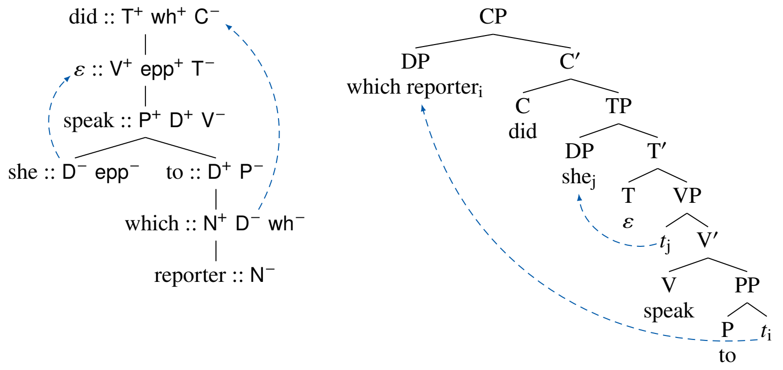


Figure 1: MG dependency tree (left) and phrase structure tree (right) for *Which reporter did she speak to?*

2 Background and model

This section briefly describes the properties of adjunction that we aim to capture, the MG dependency tree model, and SL grammars over strings, ranked trees, and unranked trees. More complex formal languages play no role in the core analysis, though several TSL and M[ulti]TSL string languages appear in 6; see Appendix A for a brief overview and example grammars.

2.1 Properties of adjunction

We are concerned primarily with the following properties of adjunction:

1. **Optionality** – an adjunct may be added or removed without affecting wellformedness
2. **Iteration** – if one adjunct may be added in some context, then any number may be added
3. **Ordering restrictions** – when two or more phrases adjoin to the same head, there may be restrictions on their order
4. **Invisibility for selection** – the properties of a phrase are determined by the those of its head, not those of any adjunct

Some simple examples of adjectival modification are provided below. (1a) demonstrates optionality and iteration: any combination of adjectives denoting size, color, and material can be used, as long as they occur in that order. The remaining examples show that other logically possible orders are degraded.

(1) a. a (big) (blue) (wooden) house
  b. ?? a blue big house
  c. ?? a wooden big house
  d. ?? a wooden blue house

Property #4 is more subtle. Empirically, it means that every phrase represented by (1a) has the same external distribution. Theoretically, it means that the features of the noun ‘house’ project, not those of the adjectives. This is easily lost in models of adjunction in which the adjective selects the noun, and can interfere with the locality of selection.

There are other properties of adjunction that we might also want to treat, but these four will be our focus, since they are directly related to the subregular complexity of adjunction. In 6.3, we will briefly touch on another structural property: the c-command paradox for right adjuncts.

2.2 MG dependency trees

Here, we briefly outline the MG (Minimalist Grammar) dependency tree model as defined in Graf and Kostyszyn (2021).[1]

In MG (Stabler, 1997, 2011), lexical items pair a phonetic exponent with a string of *features* which control how they may combine in a syntactic derivation. Standard MGs have two types of binary features, controlling the operations Merge and Move. For Merge, we have *selector* features (F^{+}) and *category* features (F^{-}). For example, the determiner *the* has features $\mathsf{N}^{+}\ \mathsf{D}^{-}$. For Move we have *licensor* features (f^{+}), marking the landing site of movement, and *licensee* features, marking the head of the mover (f^{-}). For *wh*-movement, the landing site bears wh^{+} and the mover bears wh^{-}. Additional operations require adding further feature types; we will do this for adjunction momentarily.

MGs generate a language of *derivation trees*, which encode the sequence of operations of Merge/Move/etc. Several variants exist; here we use *dependency trees*, in which all nodes are lexical

[1] The model first appears in Graf and Shafiei (2019). A nearly identical framework can also be found in Kobele (2012).

items (traditional derivation trees will be revisited in 4). Figure 1 shows a dependency tree for a simple sentence with *wh*-movement along with the corresponding phrase structure tree. The daughters of each node are its arguments, ordered by asymmetric c-command (that is, reverse order of selection). Movement is represented only via features; arrows are provided for visual convenience only.

When we say that selection is SL, we mean that licit and illicit arrangements of selector and category features can be distinguished using a SL tree grammar. Importantly, the complexity of selection itself could change if other operations are included in the tree language. We show that this does not occur in the dependency tree model: not only does adding movement not matter (as is well established) but adjunction can safely be added as well.

2.3 SL languages and grammars

SL string languages and grammars are defined in terms of *k-factors*, which are substrings of a string augmented with edge markers. For example, the 3-factors of the string *abc* are:

{⋊⋊a, ⋊ab, abc, bc⋉, c⋉⋉}

A positive SL-k grammar is a set of *permitted* k-factors, while a negative SL-k grammar is a set of *forbidden* k-factors. Here, we make use of positive grammars (interconversion is always possible). For example, a positive SL-3 grammar consisting of just the above factors would generate the string *abc* and no others. If we add the factors {cab, bca}, then we can also generate *abcabc*, *abcabcabc*, etc. By further adding {abb, bbc}, we can optionally double the *b* to produce *abbc*, *abcabbc*, etc.

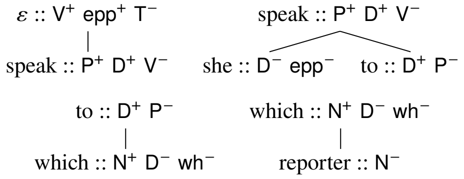
A string language is strictly k-local (SL-k) iff it can be described using a positive or negative SL-k grammar. As a regular expression, the language of the above example is $(ab(b)c)^+$. See Rogers et al. (2013) for a formal definition and further context.

SL languages/grammars are easily extended to *ranked trees*, which have a fixed maximum branching factor. They can be further extended to *unranked trees*, which have no such restriction, by associating each node with an SL string language that constrains its string of daughters.[2] We consider each of these cases in turn.

2.4 Ranked trees, selection

Traditionally, regular and subregular tree languages are defined over *ranked trees*, in which each element has a fixed number of daughters, known as its rank. The maximum branching factor of a tree is therefore bounded by the highest ranked element it contains. For such trees, a SL-k tree grammar is just a set of permitted/forbidden subtrees of height k (Rogers, 1997). For the grammar which generated the example in Figure 1, these include the following, among others:

(2) Some permitted subtrees of height 2



Of course, this can and should be condensed into a format which encodes the relevant generalizations, e.g., every verb with the selector features P^+ D^+ should have exactly two daughters, bearing D^- and P^-, in that order. We will do this in the next section. For now, we note that because the largest portion of the tree we need to examine is of height 2 and the number of possible subtrees is finite, we can list all licit/illicit subtrees, so selection is SL-2.

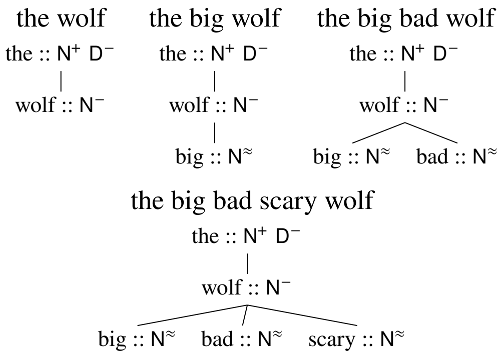
2.5 Unranked trees, adjunction

We base our system on the work of Frey and Gärtner (2002), who treat adjunction as *asymmetric feature checking*. We add a new class of *adjunction features*, notated $\mathsf{F}^{\approx}$. Modifying adjectives, for example, bear $\mathsf{N}^{\approx}$, since they adjoin to NPs. Adjunction features must be checked against a matching category feature, but the category feature of the head remains unchecked. This contrasts with Merge and Move, whose positive features must be checked against negative features in a one-to-one manner. Adjunction is therefore optional, and may also iterate.

In the MG dependency tree model, it is extremely natural to treat adjuncts as dependents of their heads, preceding all specifiers and complements. This is implicitly assumed by Shafiei and Graf (2020) in their model of adjunct islands, and I do the same in Hanson (2023a) to handle adjunct ordering. However, neither work formalizes this, nor do they treat recursive adjunction. Below are dependency trees for DPs with 0, 1, 2, and 3 NP adjuncts, respectively.

[2] Such a tree language cannot be implemented with a standard bottom-up deterministic tree automaton (BDTA). Instead, the states of the daughters are processed by a finite state string automaton, and final state of the string automaton is combined with the mother node's label to determine its state. See Comon et al. (2008) for details.

(3) Adjuncts as dependents of the head


the wolf
the :: N^+ D^-
wolf :: N^-
the big wolf
the :: N^+ D^-
wolf :: N^-
big :: $\mathsf{N}^{\approx}$
the big bad wolf
the :: N^+ D^-
wolf :: N^-
big :: $\mathsf{N}^{\approx}$
bad :: $\mathsf{N}^{\approx}$
the big bad scary wolf
the :: N^+ D^-
wolf :: N^-
big :: $\mathsf{N}^{\approx}$
bad :: $\mathsf{N}^{\approx}$
scary :: $\mathsf{N}^{\approx}$

There are two key things to notice here. First, the noun and its selector remain adjacent, as does the string of adjuncts and their head. This means that adjunction to XP is invisible to selection of XP by another head Y, as desired. Second, there is no finite bound on the number of daughters of a node. We therefore require *unranked* trees, in which the daughters of a node no longer form a tuple, but a string. Rather than exhaustively listing licit subtrees, the label of each node is mapped to a *daughter string language*, which may be infinite; many examples are given in the following sections. As for the formal implementation, the definition of an MG dependency tree language needs to be adjusted slightly, though we do not do this here.[3]

In the next section, we construct a generalized SL grammar for unranked trees which handles both selection and adjunction, and show that it works for the above structures, among others. In the following sections, we make some minor adjustments in order to incorporate recursive adjunction and adjunct ordering hierarchies.

2.6 Classes of tree grammars

The computational complexity of a tree language need not be uniform in both the vertical and horizontal dimensions. Adapting the terminology of Graf and Kostyszyn (2021), a SL-i[SL-j] tree grammar has a window of i in the vertical dimension and j in the horizontal dimension, the latter corresponding to the daughter string languages. It is also possible to use more a more powerful mechanism in one or both dimensions. For example, the analysis of movement in Graf (2022b) is TSL with a window of 2 in both dimensions, making it TSL-2[TSL-2], while the analysis of case in Hanson (2023b) is MTSL-2[TSL-2], as it involves multiple tree tiers. For present purposes, the window in the vertical dimension will never vary (it is not obvious how a window larger than size 2 would even work), but the window of the daughter string languages may vary depending on the number of arguments. When the window in the horizontal varies by daughter string language, we take the upper bound as representative.

3 Adjunction without non-locality

We begin by constructing a SL grammar which covers selection and adjunction for unranked trees, implementing the system from the previous section. We then augment the system to include recursive adjunction and adjunct ordering restrictions. The approach is closely mirrors the use of TSL tree grammars in Graf (2018) and subsequent work except that the tier projection step, needed only for long-distance dependencies, is omitted.

For now, we make no distinction between left and right adjuncts: their position in the dependency tree represents only their structural (=scopal) position. We present a potential problem with this assumption, as well as a solution, in 6.3.

3.1 Selection

First, consider the case where a node has only arguments or adjuncts among its daughters, but not both. The rules for selection and adjunction in isolation are exceedingly simple, being finite and SL, respectively. We begin with selection.

(4) **Select:** If a node bears the sequence of selector features $\mathsf{X_1}^+, \ldots, \mathsf{X_n}^+$, then its ith daughter from the right must bear category feature $\mathsf{X_i}^-$, for all $1 \leq i \leq n$.[4]

For example, *devour* is an obligatorily transitive verb, with selector features D^+ D^+. Therefore, its daughter string language consists of all strings of length two in which the category of each item is D^-. There is a finite number of selector features on any given lexical item, and the lexicon itself is finite, so the daughter string language of each node is finite, and therefore also strictly local. Specifically, if the number of arguments is n, the daughter string language is SL-$(n+1)$. In the case of *devour*:

[3] The first order constraints in Graf and Kostyszyn (2021) are meant to be combined with an appropriate axiomatization for the class of ranked finite trees; our modified version should be instead be combined with the class of unranked trees. Backofen et al. (1995) provide first-order theories of both ranked and unranked trees which are minimally different and have the desired properties, though infinite trees are not ruled out, as this requires at least monadic second order logic.

[4] Recall that the arguments of a node appear in reverse merge order.

(5) Selection grammar for *devour* (SL-3)
$G^+ = \{\rtimes\rtimes \mathsf{D}^-, \rtimes \mathsf{D}^- \mathsf{D}^-, \mathsf{D}^- \mathsf{D}^- \ltimes, \mathsf{D}^- \ltimes\ltimes\}$

The complete grammar is a map from the label of the mother to the grammar for its daughter string, based only on its selector features. If the maximum number of selector features is n, then in the classification introduced in 2.6, the complexity of the tree grammar is SL-2[SL-$(n+1)$], since we make use of a window which is of height 2 and width $(n+1)$.

3.2 Adjunction

Next, we introduce our adjunction rule.

(6) **Adjoin:** If a node bears category X^-, then it may bear zero or more daughters bearing $\mathsf{X}^{\approx}$. No other daughters with adjunction features are allowed.

For example, *wolf* bears N^-, so it may have zero or more daughters bearing $\mathsf{N}^{\approx}$. If we map the label of each node to just its adjunction feature, the daughter string language for each category X can be described with the positive grammar $\{\rtimes\ltimes, \rtimes \mathsf{X}^{\approx}, \mathsf{X}^{\approx}\mathsf{X}^{\approx}, \mathsf{X}^{\approx}\ltimes\}$, and is therefore SL-2. Since *devour* and most other verbs have at least one argument, we provide a concrete example for *wolf* instead:

(7) Adjunction grammar for *wolf* (SL-2)
$G^+ = \{\rtimes\ltimes, \rtimes \mathsf{N}^{\approx}, \mathsf{N}^{\approx}\mathsf{N}^{\approx}, \mathsf{N}^{\approx}\ltimes\}$

As stated, neither of the above rules works for nodes with both arguments and adjuncts among its daughters. Now we combine the two cases.

3.3 Combining the constraints

Recall that we assume all adjuncts to precede all arguments. Therefore, the combined daughter string language template is the concatenation of the two.

(8) **Select + Adjoin:** If a node bears the sequence of selector features $\mathsf{X_1}^+, \ldots, \mathsf{X_n}^+$ and category feature Y^-, then its daughter string consists of zero or more daughters bearing $\mathsf{Y}^{\approx}$ followed by n daughters bearing category feature $\mathsf{X_i}^-$, from right to left, for all $1 \leq i \leq n$.

SL languages are not in general closed under concatenation, so we must show that concatenation is possible in this case. Specifically, we show that the combined daughter string language schema has a factor width equal to the higher of the two source grammars: if n is the maximum number of selector features, then the combined grammar is SL-2[SL-k], where k is the greater of $\{2, (n+1)\}$.

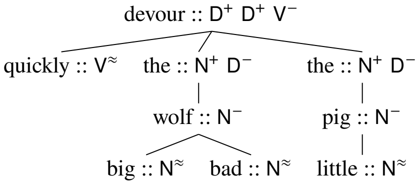
The construction is as follows. First, we convert the SL-2 adjunction grammar to SL-$(n+1)$ by padding its factors, and also remove any factors that allow a string to end without any arguments. Second, we add these to the factors of the selection grammar. Finally, we add any factors needed to transition from an adjunct to the highest argument.

A concrete example for *devour* is shown below. As before, we map each node label to just its adjunction or category feature for brevity.

(9) Combined grammar for *devour* (SL-3)
$G^+ = \{\rtimes\rtimes \mathsf{D}^-, \rtimes \mathsf{D}^- \mathsf{D}^-, \mathsf{D}^- \mathsf{D}^- \ltimes, \mathsf{D}^- \ltimes\ltimes, \rtimes\rtimes \mathsf{V}^{\approx}, \rtimes \mathsf{V}^{\approx}\mathsf{V}^{\approx}, \mathsf{V}^{\approx}\mathsf{V}^{\approx}\mathsf{V}^{\approx}, \rtimes \mathsf{V}^{\approx}\mathsf{D}^-, \mathsf{V}^{\approx}\mathsf{V}^{\approx}\mathsf{D}^-, \mathsf{V}^{\approx}\mathsf{D}^-\mathsf{D}^-\}$

Let us apply this grammar to the node *devour* in the dependency tree for the sentence *The big bad wolf quickly devoured the little pig*, shown below. For simplicity, we truncate the tree at the VP level and omit movement features. The reader may confirm that all 3-factors of the daughters of *devour* are licit. To ensure that the entire tree is licit, we repeat this procedure for every node.

(10) a. Dependency tree:



b. DS of *devour*: $\mathsf{V}^{\approx}\ \mathsf{D}^-\ \mathsf{D}^-$

c. 3-factors of DS: $\{\rtimes\rtimes \mathsf{V}^{\approx}, \rtimes \mathsf{V}^{\approx}\mathsf{D}^-, \mathsf{V}^{\approx}\mathsf{D}^-\mathsf{D}^-, \mathsf{D}^-\mathsf{D}^-\ltimes, \mathsf{D}^-\ltimes\ltimes\}$

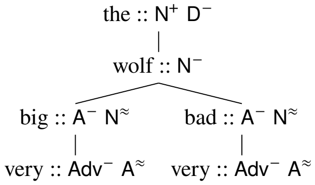
The construction is essentially identical for items with three or more arguments. For those with just one, the selection grammar is already SL-2, so no padding of the adjunction factors is required. For items with no arguments (including the verb *rain* and many nouns), we are back to the plain adjunction grammar, which remains SL-2.

To briefly review, we achieved a combined SL model of selection and adjunction over unranked trees, whose grammar is a mapping from node labels to daughter string languages, each of which is SL, for a combined complexity of SL-2[SL-k], with $k \geq 2$. Now, we introduce recursive adjunction.

4 Recursive adjunction

We follow the lead of Graf (2018) by reintroducing category features on adjuncts. For example, modifying adjectives carry $\mathsf{A}^-\ \mathsf{N}^{\approx}$, and adverbs carry either $\mathsf{Adv}^{-}\ \mathsf{A}^{\approx}$ or $\mathsf{Adv}^{-}\ \mathsf{V}^{\approx}$. Below is an example adverbial modification of adjectives, which in turn modify a noun.

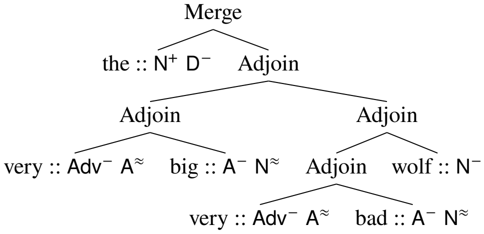
(11) the very big very bad wolf



Locality is clearly preserved in the dependency tree model, since adding an adverb under an adjective does not interrupt adjacency between the adjective and the head noun, just as adding an adjective below a noun does not affect the relation with the selecting determiner. Furthermore, although some category features are no longer checked with a corresponding selector feature, this can be determined just from the label of the node in question, so we do not even need to change the SL tree grammar. We continue to distinguish items with and without a final adjunction feature, just as before.

At this point, I should briefly describe the problem that occurs with recursive adjunction in traditional MG derivation trees. In this system, internal nodes represent the Merge/Move/Adjoin operations, and all leaves are lexical items. The derivation tree for the current example is shown below.

(12) the very big very bad wolf



Here, an adjunct and its head are not necessarily adjacent, and the distance grows without bound if the adjunct itself serves an an adjunction site. As a consequence, adjunction is not SL for any window size. Furthermore, selection is not SL either, since the distance between the D head and the N head grows without bound as adjuncts are added. If not for recursive adjunction, strict locality of selection could be rescued via a chain analysis (e.g. D licenses A, which licenses A, which licenses N). But with recursive adjunction, the intervening A heads themselves are not guaranteed to lie within any finite window.

According to Graf (2018), Merge and Move are *structure-sensitive TSL* (SS-TSL) over derivation trees (see De Santo and Graf 2019 for the string case); the complexity of Adjoin is left open, though it is clearly not SL. Several phonological phenomena are SS-TSL over strings (Graf and Mayer, 2018; Mayer and Major, 2018), so this is not a catastrophe. Additionally, as Graf notes, it would mean that Merge and Move are extremely closely related in formal terms, mirroring the view in Chomsky (2004). However, as the evidence accumulates that SL and TSL are sufficient for most syntactic phenomena under the dependency tree model (Graf, 2022b; Hanson, 2023b, 2025), one gets the impression that the need for SS-TSL is an artifact of the derivation tree representation.

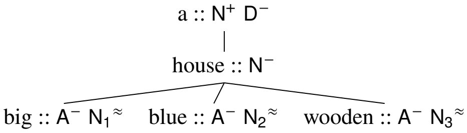
This requires elaboration since, as a reviewer remarks, there is an inherent trade-off between representational and computational complexity, such that one can often be reduced by increasing the other. In this case, the information in each representation is comparable, with sister order in the dependency replacing the extra nodes of the derivation tree, but the computational complexity of the former is lower. Furthermore, the range of patterns which SL/TSL can produce have wide empirical support, while SS-TSL serves primarily to factor out the extra nodes of the derivation tree. An exception can be found in Principle B of the binding theory, which seems to require SS-TSL (Graf and Shafiei, 2019), mirroring the occasional SS-TSL pattern in phonology, but this does not seem to be needed for most operations. In summary, the dependency tree model allows us to minimize the overall complexity of the system while also providing the best fit to the known typology.

5 Adjunct ordering restrictions

The adjustment we made for recursive adjunction also lays the groundwork for encoding adjunct ordering restrictions. The basic insight by Fowlie (2013) is that a principled treatment of adjunction ordering requires a *pair of features* rather than a single adjunction feature. By tracking both the position in the hierarchy and the adjunction target simultaneously, we can avoid resorting to low level tricks such as adding unmotivated empty categories or exploding the lexicon.

Rather than implementing her exact system, we make use of the pairing of category and adjunction features already in play. Specifically, we split our adjunction features by adding an index corresponding to the position the item must take in the relevant ordering hierarchy. The primary difference between our approach and Fowlie's is that while she uses paired features primarily to label adjunction nodes in the derivation tree, we label the adjuncts themselves. Consider again the example from (1), repeated below with its dependency tree.

(13) a big blue wooden house



In this case, we included only three indices, but we can include as many as we need as long as the number of positions is finite. In the style of the preceding examples, the rule is as follows:

(14) **Ordered adjunction:** If a node bears category feature X which has n positions in its adjunction hierarchy, then any pair of daughters $\mathrm{d_i}$, $\mathrm{d_j}$ bearing $\mathsf{X_i}^{\approx}$ and $\mathsf{X_j}^{\approx}$, where $1 \leq i < j \leq n$, must be ordered such that $\mathrm{d_i}$ precedes $\mathrm{d_j}$.

As discussed by Hanson (2023a), ordered adjunction is SL-2, just like simple adjunction, even allowing for iteration, e.g. *the big big big blue house*. Viewed as a finite state automaton, the daughter string language is just the reflexive transitive closure of the order of adjunction categories. Rather than clutter the above definition, we proceed directly to the template which covers all cases:

(15) Adjunction grammar for category N (SL-2)
$G^{+} = \{\rtimes\ltimes,\ \rtimes\mathsf{N_1}^{\approx},\ \rtimes\mathsf{N_2}^{\approx},\ \rtimes\mathsf{N_3}^{\approx},\ \mathsf{N_1}^{\approx}\mathsf{N_2}^{\approx},\ \mathsf{N_2}^{\approx}\mathsf{N_3}^{\approx},\ \mathsf{N_1}^{\approx}\mathsf{N_1}^{\approx},\ \mathsf{N_2}^{\approx}\mathsf{N_2}^{\approx},\ \mathsf{N_3}^{\approx}\mathsf{N_3}^{\approx},\ \mathsf{N_1}^{\approx}\ltimes,\ \mathsf{N_2}^{\approx}\ltimes,\ \mathsf{N_3}^{\approx}\ltimes\}$

This grammar can then be combined with the selection grammar as before.

It is natural to ask whether it might be better to split the category feature of the adjunct rather than the adjunction feature. This would also presumably work, and would in fact be more faithful to Fowlie's system. One small downside is that increases the size of the lexicon somewhat. For example, if we split category A into S(ize)/C(olor)/M(aterial)/etc. then predicational use of adjectives will require duplicate lexical entries for all selecting heads (*be*, *seem*, etc.). The same is true of adjective modifiers such as *very*. While Fowlie presents some potential solutions, the present approach sidesteps these problems altogether, as the effects of the split are isolated to just the context where they are desired. We further discuss this alternative in Section 6.4.

6 Extensions and alternatives

At this point, we have achieved what we set out to do: we have constructed a simple SL model of adjunction which handles all of the properties specified at the outset, and which avoids non-locality in cases of recursive adjunction. Now, we address some other issues which have not been our focus, some possible extensions of the current system, and how some other systems compare. For brevity, some of the string languages in this section are defined using regular expressions, with the grammars relegated to Appendix A.

6.1 Ordered and unordered adjuncts

So far, we have mostly ignored adjuncts without ordering hierarchies, which traditionally include PPs. To a certain degree, there is not much to say about them since, if they are indeed unordered with respect to each other, then the simple adjunction grammar from Section 3 will do the job. The fact that they are linearized to the right in English can be seen as a part of the mapping to the surface, independent of the dependency tree.

However, there is potential danger to the SL analysis when we consider both ordered and unordered adjuncts together. Suppose for the sake of argument that PPs can be interspersed among adjectives or adverbs (as determined by scope), and that they can also iterate in each position. This would yield a daughter string language along the following lines:

(16) Ordered APs and unordered PPs
$P^{*} \cdot A_{1}^{*} \cdot P^{*} \cdot A_{2}^{*} \cdot P^{*} \cdot A_{3}^{*} \cdot P^{*}$

Such a language is not SL, since we need adjacent items in the adjective hierarchy to appear in the same window yet there is no limit to the number of P heads which may intervene. It is uncertain whether this scenario is actually realistic, but if so, then the daughter string languages for selection and each type of adjunction become TSL, and the combined language would be Multi-TSL (MTSL; see De Santo and Graf 2019), since the tiers for each would be different. Even if such constructions exist, it could be that left and right adjuncts are not actually interspersed in the dependency tree, in which the daughter string language remains SL-2. We consider this possibility in Section 6.3.

6.2 The position of adjunction

In the above analysis, we assumed that all adjuncts precede all arguments in the derivation tree, which is equivalent to the assumption that all adjunction occurs at the XP level. It is also conceivable that adjuncts could occur in other positions. For example, Frey and Gärtner (2002) assume that manner adverbs attach to the verb before the object does in their analysis of German.

We should therefore consider the possibility that the position of adjunction features within the MG feature string may vary. Indeed, one could make the argument that the SL model predicts such variation. In the example just cited, all manner adverbs follow the complement, which is an easy change. We might also ask whether there exist any systems which are not SL. For example, consider a hypothetical language in which PP adjuncts can be inserted freely in any position, similar to (16):

(17) Hypothetical non-SL version of *devour*
$P^* \cdot D \cdot P^* \cdot D \cdot P^*$

This particular example would again be MTSL. If adjunction is SL, then such adjunction paradigms should not exist, even if some other variants do.

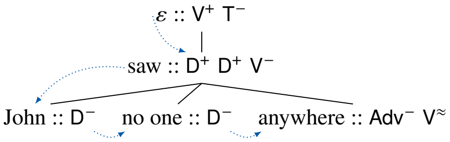
6.3 Left vs. right adjuncts

Right adjuncts in English are unordered, with constituency and scope diagnostics suggesting that the outer adjuncts are higher, but c-command diagnostics such as NPI licensing go the other way.

(18) a. John saw [no one] [anywhere].
b. * John saw [anyone] [nowhere].
(Ernst, 1994)

In previous work (Graf and Shafiei, 2019; Hanson, 2025), a relation called *d[erivational]-command*, which combines the dominance and left-sister relations of the dependency tree, serves as the analog of c-command in the phrase structure tree. The NPI data can therefore be accommodated if we assume that right adjuncts appear *after* all arguments in the dependency tree.

(19) Abbreviated dependency tree for (18a), showing d-command relations



In doing so, we affirm the idea that sister order encodes command at the expense of losing a direct correspondence to constituency and scope. These would instead need to be introduced in the mapping from the dependency tree to the corresponding phrase structure tree. Such an approach would be reminiscent of the dual model of 'cascade syntax' and 'layered syntax' in Pesetsky (1996). I leave the exploration of this possibility to future work.

6.4 Adjunct subcategories

As mentioned in Section 5, the closest alternative to the proposed approach to adjunct ordering—splitting the category rather than the adjunction feature—introduces some lexical redundancy independent of that introduced by the inclusion of adjunction features. But perhaps we could do away with adjunction features entirely and rely on the local context to identify adjunction, as in Fowlie (2013). The structure of the daughter string language would be essentially identical, just with $N_1^{\approx}/N_2^{\approx}/N_3^{\approx}$ substituted by $S^-/C^-/M^-$, and so on. This has been done for example (15) below:

(20) Adjunction grammar for category N (SL-2)
$G^+ = \{\rtimes\ltimes, \rtimes S^-, \rtimes C^-, \rtimes M^-, S^-C^-, C^-M^-,$
$S^-S^-, C^-C^-, M^-M^-, S^-\ltimes, C^-\ltimes, M^-\ltimes\}$

Aside from creating some lexical redundancy in the selectors of S/C/M/etc., a major disadvantage of such a model from a subregular perspective is that arguments and adjuncts of the same category can no longer be easily distinguished for long-distance operations such as movement, as sisterhood in the dependency tree is not preserved by projection to a tree tier. Arguments and adjuncts are usually thought to differ in their behavior with respect to movement (both as movers and as containers for movers), casting doubt on the viability of such an approach, though see 6.6 for a counterargument.

6.5 Selectional approaches

As noted by Fowlie (2013), models that attempt to reduce adjunction to selection suffer from various formal and linguistic shortcomings, particularly in accounting for ordering hierarchies. For example, we could implement a functional sequence, e.g. D < S < C < M < N, by including empty elements to fill the unused slots. Each modifier needs a single lexical entry, but the empty items have no independent morphological or semantic motivation and are therefore "nothing more than a trick to hold the syntax together" (Fowlie, 2013, p. 16).

(21) Functional sequence

wooden :: $\mathsf{N}^+\ \mathsf{M}^-$ ε :: $\mathsf{N}^+\ \mathsf{M}^-$
blue :: $\mathsf{M}^+\ \mathsf{C}^-$ ε :: $\mathsf{M}^+\ \mathsf{C}^-$
big :: $\mathsf{C}^+\ \mathsf{S}^-$ ε :: $\mathsf{C}^+\ \mathsf{S}^-$
the :: $\mathsf{S}^+\ \mathsf{D}^-$

Conversely, we can avoid empty heads by means of lexical homophony, but the lexical redundancy factor is far worse than other alternatives, on the order of n^2 (ibid.). Even if we dismiss the increased memory burden, the pattern feels particularly accidental when analyzed in this way, as there is nothing which prevents items of the same category from selecting a different set of 'next' elements.

(22) Massive homophony

wooden → $\mathsf{N}^+\ \mathsf{M}^-$
blue → $\mathsf{N}^+\ \mathsf{C}^-$ / $\mathsf{M}^+\ \mathsf{C}^-$
big → $\mathsf{N}^+\ \mathsf{S}^-$ / $\mathsf{M}^+\ \mathsf{S}^-$ / $\mathsf{C}^+\ \mathsf{S}^-$
the → $\mathsf{N}^+\ \mathsf{D}^-$ / $\mathsf{M}^+\ \mathsf{D}^-$ / $\mathsf{C}^+\ \mathsf{D}^-$ / $\mathsf{S}^+\ \mathsf{D}^-$

A third alternative, not considered by Fowlie, utilizes 'adjunctizer' heads which introduce the adjunct itself as a specifier. This contains the scope of redundancy to a small subset of the lexicon, but then we are back to the problem of unmotivated empty elements.

(23) Adjunctizer heads

M-ADJ → $\mathsf{N}^+\ \mathsf{M}^+\ \mathsf{M}^-$
C-ADJ → $\mathsf{N}^+\ \mathsf{C}^+\ \mathsf{C}^-$ / $\mathsf{M}^+\ \mathsf{C}^+\ \mathsf{C}^-$
S-ADJ → $\mathsf{N}^+\ \mathsf{S}^+\ \mathsf{S}^-$ / $\mathsf{M}^+\ \mathsf{S}^+\ \mathsf{S}^-$ / $\mathsf{C}^+\ \mathsf{S}^+\ \mathsf{S}^-$
the → $\mathsf{N}^+\ \mathsf{D}^-$ / $\mathsf{M}^+\ \mathsf{D}^-$ / $\mathsf{C}^+\ \mathsf{D}^-$ / $\mathsf{S}^+\ \mathsf{D}^-$

In each case, the difficulty of distinguishing arguments and adjuncts which we noted earlier still applies. Overall, it seems to be preferable to keep adjunction as a distinct operation, and factor out ordering restrictions into the SL grammar.

6.6 Additional puzzles

Throughout this paper, I have assumed that the given generalizations about adjunction are actually correct, but various exceptions have long been known. For example, as a reviewer notes, violations of the adjective order in cases of recursive adjunction seem less bad compared to simple adjunction.

(24) a. a big blue house
b. ?? a blue big house

(25) a. a very big very blue house
b. ? a very blue very big house

As discussed by Hanson (2023a), there are various ways in which adjunct orders are more fluid than is often supposed; in languages such as German, they seem not to exist (Thomas Graf, p.c.). It is therefore not clear that they should even be modeled in the syntactic grammar. For present purposes, the crucial point is that if we decide to do so, they remain within the power of SL.

Similarly, I have taken for granted that the argument-adjunct distinction exists and must be accounted for. This might also not be so clear cut: a reviewer cites McInnerney (2022), who argues that the distinction is not well-supported on syntactic or semantic grounds. This seems compatible with the central claim of this paper, since selection and adjunction are SL both in isolation and in combination. It would be only a small step to eliminate the distinction entirely, with the caveats discussed in 6.5. That said, given that the study by McInnerney focuses almost exclusively on PPs, further investigation is needed to determine whether the same arguments apply to adjectives and adverbs.

7 Conclusion

I have shown that a linguistically appealing model of adjunction based on a pairing of category and adjunction features is SL over MG dependency trees, inclusive of formal challenges such as recursive adjunction and adjunct ordering restrictions. Selection and adjunction can be combined into a single SL daughter string language, and beyond this, certain variants such as low manner adverb attachment and the distinction between left and right adjuncts may be accommodated.

Overall, these results support the classification of adjunction as a local phenomenon. If it is determined that the interspersing of ordered and unordered adjuncts in the dependency tree cannot be avoided, then the combined complexity of selection and adjunction increases to SL-2[MTSL-k]. Now that most major syntactic operations (selection, adjunction, movement, case, agreement, binding) have been studied in isolation, the next step is to determine to what extent the interactions between them can be handled within the bounds of the (M)TSL tree languages.

Acknowledgments

This work was partly supported by the Institute for Advanced Computational Science at Stony Brook University. My thanks to Thomas Graf and three anonymous reviewers for their detailed feedback on earlier versions of this paper.

References

Rolf Backofen, James Rogers, and K. Vijay-Shanker. 1995. A first-order axiomatization of the theory of finite trees. *Journal of Logic, Language and Information*, 4(1):5–39.

Noam Chomsky. 2004. Beyond explanatory adequacy. In Adriana Belletti, editor, *Structures and Beyond*, pages 104–131. Oxford University Press, New York, NY.

Hubert Comon, Max Dauchet, Rémi Gilleron, Florent Jacquemard, Denis Lugiez, Christof Löding, Sophie Tison, and Marc Tommasi. 2008. Tree automata techniques and applications. Online.

Aniello De Santo and Thomas Graf. 2019. Structure sensitive tier projection: Applications and formal properties. In *Formal Grammar: 24th International Conference*, pages 35–50, Riga, Latvia.

Thomas Ernst. 1994. M-command and precedence. *Linguistic Inquiry*, 25(2):327–335.

Meaghan Fowlie. 2013. Order and optionality: Minimalist Grammars with adjunction. In *Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13)*, pages 12–20, Sofia, Bulgaria.

Werner Frey and Hans-Martin Gärtner. 2002. On the treatment of scrambling and adjunction in minimalist grammars. In *Proceedings of Formal Grammar 2002*, pages 41–52, Trento, Italy.

Thomas Graf. 2014. Models of adjunction in minimalist grammars. In *Formal Grammar: 19th International Conference*, pages 52–68, Tübingen, Germany.

Thomas Graf. 2018. Why movement comes for free once you have adjunction. In *Proceedings of CLS 53*, pages 117–136, Chicago, IL.

Thomas Graf. 2022a. Subregular linguistics: bridging theoretical linguistics and formal grammar. *Theoretical Linguistics*, 48(3–4):145–184.

Thomas Graf. 2022b. Typological implications of tier-based strictly local movement. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2022*, pages 184–193, Online.

Thomas Graf and Kalina Kostyszyn. 2021. Multiple wh-movement is not special: The subregular complexity of persistent features in Minimalist Grammars. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2021*, pages 275–285, Online.

Thomas Graf and Connor Mayer. 2018. Sanskrit n-retroflexion is input-output tier-based strictly local. In *Proceedings of SIGMORPHON 2018*, pages 151–160, Brussels.

Thomas Graf and Nazila Shafiei. 2019. C-command dependencies as TSL string constraints. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 205–215, New York, NY.

Kenneth Hanson. 2023a. Strict locality in syntax. In *Proceedings of CLS 59*, pages 131–145, Chicago, IL.

Kenneth Hanson. 2023b. A TSL analysis of Japanese case. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2023*, pages 15–24, Amherst, MA.

Kenneth Hanson. 2025. Tier-based strict locality and the typology of agreement. To appear in *Journal of Language Modelling*.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. In Larry M. Hyman and Frans Plank, editors, *Phonological Typology*, pages 126–195. De Gruyter Mouton.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 58–64, Portland, OR.

Gregory M. Kobele. 2012. Eliding the derivation: A minimalist formalization of ellipsis. In *Proceedings of the 19th International Conference on Head-Driven Phrase Structure Grammar*, Chungnam National University Daejeon.

Dakotah Lambert and James Rogers. 2020. Tier-based strictly local stringsets: Perspectives from model and automata theory. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2020*, pages 159–166, New Orleans, LA.

Connor Mayer and Travis Major. 2018. A challenge for tier-based strict locality from uyghur backness harmony. In *Formal Grammar 2018: 23rd International Conference*, pages 62–83, Sofia, Bulgaria.

Andrew McInnerney. 2022. *The Argument/Adjunct Distinction and the Structure of Prepositional Phrases*. Ph.D. thesis, University of Michigan, Ann Arbor, MI.

David Pesetsky. 1996. *Zero syntax: Experiencers and cascades*. MIT Press, Cambridge, MA.

James Rogers. 1997. Strict LT_2 : Regular :: Local : Recognizable. In *Logical Aspects of Computational Linguistics: First International Conference, LACL '96 (Selected Papers)*, pages 366–385, Nancy, France.

James Rogers, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel. 2013. Cognitive and sub-regular complexity. In *Formal Grammar: 17th and 18th International Conferences*, pages 90–108.

Nazila Shafiei and Thomas Graf. 2020. The subregular complexity of syntactic islands. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2020*, pages 421–430, New Orleans, LA.

Edward P. Stabler. 1997. Derivational minimalism. In *Logical Aspects of Computational Linguistics: First International Conference, LACL '96 (Selected Papers)*, pages 68–95, Nancy, France.

Edward P. Stabler. 2011. Computational perspectives on Minimalism. In Cedric Boeckx, editor, *Oxford handbook of linguistic Minimalism*, pages 617–643. Oxford University Press, Oxford, UK.

Mai Ha Vu, Nazila Shafiei, and Thomas Graf. 2019. Case assignment in TSL syntax: A case study. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 267–276, New York, NY.

A Additional adjunction grammars

Section 6 provided examples of daughter string languages for several hypothetical adjunction patterns, not all of which are SL. Very briefly, a TSL language is one in which certain elements are ignored, forming a *tier projection*. Which elements appear on the tier is determined completely by their labels, and those that do are treated as if adjacent, subject to a SL grammar. See Heinz et al. (2011); Lambert and Rogers (2020) for details. An MTSL grammar is just the intersection of several TSL grammars (De Santo and Graf, 2019).

A.1 Ordered and unordered adjuncts

First, the hypothetical language from 6.1, which freely intersperses ordered AP adjuncts and unordered PP adjuncts, is repeated below. The constraints of the grammars are unchanged from our earlier SL grammars. The only difference is that tier projection is used to ignore adjuncts of the opposite type. For simplicity, I use mnemonic labels rather than MG feature specifications.

(26) Ordered APs and unordered PPs (MTSL-2)

a. Language:
$P^* \cdot A_1^* \cdot P^* \cdot A_2^* \cdot P^* \cdot A_3^* \cdot P^*$

b. AP adjunction grammar (TSL-2)
$T = \{A_1, A_2, A_3\}$

$$G^+ = \left\{ \begin{array}{c} \rtimes\ltimes, \rtimes A_1,\ \rtimes A_2,\ \rtimes A_3, \\ A_1A_2,\ A_2A_3, \\ A_1A_1,\ A_2A_2,\ A_3A_3, \\ A_1\ltimes,\ A_2\ltimes,\ A_3\ltimes \end{array} \right\}$$

c. PP adjunction grammar (TSL-2)
$T = \{P\}$
$G^+ = \{\rtimes\ltimes, \rtimes P, PP, P\ltimes\}$

A.2 Unordered adjunction everywhere

If unordered adjuncts can be freely interspersed with arguments, the result is MTSL, similar to free mixing of ordered and unordered adjuncts. In Section 6.2, I predicted that this should not occur.

(27) Unordered adjunction + selection (MTSL-3)

a. Language:
$P^* \cdot D \cdot P^* \cdot D \cdot P^*$

b. Selection grammar (TSL-3)
$T = \{D\}$
$G^+ = \{\rtimes\rtimes D, \rtimes DD, DD\ltimes, D\ltimes\ltimes\}$

c. Adjunction grammar (TSL-2)
$T = \{P\}$
$G^+ = \{\rtimes\ltimes, \rtimes P, PP, P\ltimes\}$

A.3 Low adjunction

The proposed daughter string language and grammar proposed for low manner adverbs as described in 6.2 is given below. Unlike the previous grammars, this one remains SL.

(28) Low adjunction equivalent of *devour* (SL-3)

a. Language:
$D \cdot D \cdot Adv^*$

b. Grammar:

$$G^+ = \left\{ \begin{array}{c} \rtimes\rtimes D,\ \rtimes DD,\ DD\ltimes, \\ D\ltimes\ltimes,\ D\,D\,Adv,\ D\,Adv\,Adv, \\ Adv\,Adv\,Adv,\ D\,Adv\ltimes, \\ Adv\,Adv\ltimes,\ Adv\ltimes\ltimes \end{array} \right\}$$

This could be further generalized to allow different types of adjuncts in different positions as long as they can be distinguished from one another, as shown below.

A.4 Left and right adjuncts

I noted in 6.3 that a grammar with left adjuncts at the beginning and right adjuncts at the end would be SL, as long as distinct indices are used. In this case, we can safely allow ordered adverbs on the left and unordered adverbs and PPs on the right.

For simplicity, I assume a single index R for right adjuncts, and I do not pad the 2-factors to 3-factors as is technically required (such a factor should be interpreted as standing in for any 3-factor that contains it as a substring). Effectively, we combine the grammars from (9) and (28).

(29) Left and right adjunction (SL-3)

a. Language:
$Adv_1^* \cdot Adv_2^* \cdot Adv_2^* \cdot D \cdot D \cdot Adv_R^*$

b. Grammar:

$$G^+ = \left\{ \begin{array}{c} \rtimes\ltimes,\ \rtimes A_1,\ \rtimes A_2,\ \rtimes A_3, \\ A_1A_2,\ A_2A_3, \\ A_1A_1,\ A_2A_2,\ A_3A_3, \\ A_1D,\ A_2D,\ A_3D, \\ \rtimes\rtimes D,\ \rtimes DD,\ DD\ltimes, D\ltimes\ltimes, \\ DA_R, A_RA_R, A_R\ltimes \end{array} \right\}$$

Reconciling categorical and gradient models of phonotactics

Connor Mayer
University of California, Irvine
cjmayer@uci.edu

Abstract

Should phonotactic knowledge be modeled as categorical or gradient? In this paper, I present new data from a Turkish acceptability judgment study that addresses some limitations of previous work on this question. This study shows that gradient models account for the variability in acceptability ratings better than categorical ones. However, I suggest that the distinction between gradient and categorical models is somewhat superficial when we think of models in a mathematically general way. I propose on this basis that both categorical and gradient models have a role to play in linguistic research.

1 Is phonotactics gradient or categorical?

Phonotactics is the restrictions that languages place on how sounds can be sequenced into words. Different languages impose different phonotactic restrictions. For example, although English and Spanish both contain the sounds {k, p, s, i}, a word like /skip/ 'skeep' is only possible in English. Spanish has more restrictive phonotactics, prohibiting /s/-initial complex onsets. For similar reasons, a word like /fstʃɔŋs/ is a perfectly fine Polish word (*wstrząs* 'shock'), but would not be a suitable English word because of English's more restrictive onset phonotactics. It is generally accepted that phonotactic knowledge is learned by generalizing across forms in the lexicon (e.g. Chomsky and Halle, 1968; Bailey and Hahn, 2001; Edwards et al., 2004).

One common method of probing phonotactic knowledge is phonotactic acceptability judgments, where participants are asked to rate the acceptability of novel words as possible words in their language. A longstanding empirical observation is that phonotactic acceptability judgments are *gradient*. That is, participants do not simply treat words as acceptable or not, but rather ascribe varying degrees of acceptability to them. A classic example from Chomsky and Halle (1968) is the three nonce words /blɪk/, /bnɪk/, and /bnzk/. Despite all three being unattested in English, English speakers (or at least Chomsky and Halle) rank them in terms of acceptability such that /bnzk/ $\ll$ /bnɪk/ $\ll$ /blɪk/. That is, speakers judge /bnɪk/ to be a more acceptable word than /bnzk/, but a less acceptable word than /blɪk/. Similar results have been found in a wide range of studies (e.g. Coleman and Pierrehumbert, 1997; Scholes, 1966; Hayes, 2000; Bailey and Hahn, 2001; Hayes and Wilson, 2008; Albright, 2009; Daland et al., 2011, a.o.).

Two question that naturally arise from these results are where this gradience comes from and how we should represent it in our models of language. There have been two broad theoretical approaches, which we will cover in the following sections (see Schütze, 1996, for a discussion of these perspectives in linguistics more broadly).

1.1 Gradient models of phonotactics

The first approach proposes that we see gradience in these studies because the phonotactic grammar is itself gradient, or that a gradient measure of acceptability can be derived from the grammar. Chomsky and Halle (1968) write that "a real solution to the problem of 'admissibility' will not simply define a tripartite categorization of occurring, accidental gap, and inadmissible, but will define the 'degree of admissibility' of each potential lexical matrix in such a way as to distinguish /blɪk/ from /bnɪk/ and /bnɪk/ from /bnzk/, and to make numerous other distinctions of this sort" (pp. 416–417). They operationalize this 'degree of admissibility' as a quantity derived from the phonological grammar and the lexicon: the minimum number of featural changes required to convert a word into an existing word in the language. Chomsky and Halle also note that this gradience exists within the lexicon itself (p. 418). In English, for example, there are semi-admissible words like /sfɪŋks/ 'Sphinx' that constitute exceptions to otherwise strong phonotactic restrictions on onset formation.

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 144-154.
Eugene, Oregon, July 18-20, 2025

Chomsky and Halle do not do away with the concept of grammaticality: there are still forms that can be produced by the grammar and forms that cannot. Rather, they suggest that a gradient acceptability score can be derived from the grammar by some additional mechanism. Subsequent proposals have gone further, claiming that the grammar itself generates both categorical and gradient outcomes: whether we get one or the other depends primarily on the amount of variability in the learning data. It's beyond the scope of this paper to cover these approaches in detail, but many have been expressed within the context of Optimality Theory (Prince and Smolensky, 1993/2004) and typically either vary constraint rankings in order to generate gradient outcomes (e.g. Hayes, 2000) or derive probabilities from weighted constraints (e.g. Hayes and Wilson, 2008; Dai et al., 2023). Gradient models of phonotactics have also been proposed in the context of formal language theory (Mayer, 2021). Under these approaches, gradience emerges from an interaction between the grammar and the learning data, not a bespoke mechanism.

This perspective is supported outside the world of generative linguistics, where phonotactic knowledge is typically treated as gradient, and is often represented by simple probabilistic n-gram models (Markov, 1913; Shannon, 1948). Gradient knowledge of phonotactics has been claimed to play an important role in areas such as speech perception (e.g. Norris and McQueen, 2008; Dupoux et al., 2011; Chodroff and Wilson, 2014; Steffman and Sundara, 2023), speech production (e.g. Edwards et al., 2004), word segmentation and learning (e.g. Mattys et al., 1999; McQueen, 1998; Mersad and Nazzi, 2011; Vitevitch and Luce, 1999; Storkel, 2001), and speech errors (e.g. Goldrick and Larson, 2008; Taylor and Houghton, 2005; Warker, 2013; Warker and Dell, 2006, 2015), among others.[1]

1.2 Categorical models of phonotactics

The second theoretical approach to gradience proposes that the phonotactic grammar is fundamentally categorical (that is, it really does judge words to be acceptable or not) and that gradience in acceptability judgments is solely the result of extra-grammatical factors such as task effects or misperception (e.g. Gorman, 2013; Durvasula, 2020; Kostyszyn and Heinz, 2022; Dai, 2025). There are two main sources of evidence for this view.

The first is that extra-grammatical performance factors have indeed been shown to influence phonotactic judgments. A convincing demonstration of this comes from Kahng and Durvasula (2023), who show that some variability in nonce word judgments by Korean speakers is the result of misperception of certain consonant clusters.

The second source of evidence is several studies suggesting that categorical models do as well as or better than gradient models in predicting acceptability judgments. As Gorman (2013) puts it, "simple baselines better account for gradient well-formedness judgements than current computational models of phonotactic knowledge, suggesting that the gradience observed in these tasks [does] not derive from known grammatical mechanisms" (p. 17). Specifically, categorical models have been claimed to better predict English onset acceptability (Gorman, 2013; Durvasula, 2020; Dai, 2025), Polish onset acceptability (Kostyszyn and Heinz, 2022; Dai, 2025), Turkish vowel harmony (Gorman, 2013; Dai, 2025) and English medial consonant cluster distributions (Gorman, 2013).

We will focus on the second type of evidence here. With regards to the first, note that proponents of gradient models do not suggest that extra-grammatical factors have no role at all in the gradience exhibited in acceptability judgment tasks. Rather, the claim is that a substantial part of the gradience can be predicted by grammatical factors. Hayes (2000) puts it as follows:

> [P]atterns of gradient well-formedness often seem to be driven by the very same principles that govern absolute well-formedness [...] I conclude that the proposed attribution of gradient well-formedness judgments to performance mechanisms would be uninsightful. Whatever "performance" mechanisms we adopted would look startlingly like the grammatical mechanisms that account for non-gradient judgments (p. 90).

In other words, gradience in acceptability studies is often predictable from "soft" versions of the same constraints that govern more categorical patterns like phonological alternations.

[1] We do not consider neighborhood density here, another important property that influences wordlikeness judgments. For discussion of the relationship between neighborhood density and phonotactic probability, see e.g. Bailey and Hahn (2001); Steffman and Sundara (2024).

1.3 Limitations of past work

There are three important limitations to previous work comparing categorical and gradient models of phonotactics. First, these papers have used a relatively small number of data sets, almost all focusing on consonant clusters. This makes it difficult to evaluate how generally these results hold across different types of phonotactic dependencies.

The second limitation is that the authors of these papers do not all subscribe to the same definition of categorical. In some cases the grammar truly is categorical, assigning words either grammatical or ungrammatical status (Gorman, 2013; Kostyszyn and Heinz, 2022; Dai, 2025). In other cases, similar to Chomsky and Halle (1968), some secondary gradient measure of admissibility is derived from a categorical grammar (Durvasula, 2020; Kostyszyn and Heinz, 2022). We will treat these two definitions of categorical as separate models below.

The third limitation is that the gradient model typically used is the UCLA Phonotactic Learner (Hayes and Wilson, 2008), an influential phonotactic learning model implemented in the maximum entropy Optimality Theory framework (Goldwater and Johnson, 2003; Mayer et al., 2024). Although it does implement a gradient model of phonotactics, it has the additional task of inducing the constraints themselves from the data. The categorical models in these papers are typically provided with predefined constraints (though cf. Dai, 2025). It is unclear whether the poor performance of the UCLA learner is due to the fact that it is gradient or to some aspect of the constraint induction process. The UCLA learner is also sensitive to how it is parameterized, and it is not typical for these studies to compare performance under a range of hyperparameters.

1.4 The remainder of the paper

While this paper will by no means resolve this debate, I will try to achieve two more modest goals. First, I will present new data from a phonotactic acceptability judgment study of Turkish that addresses some of the limitations expressed above. This study will show that gradient models are better able to predict participant judgments. Second, I will try to convince you that the distinction between categorical and gradient grammars is in fact a somewhat superficial one when we consider the matter from a mathematical perspective, and that both conceptualizations of the grammar have a role to play in linguistic research and theory-building.

2 Defining our grammars

We will consider three classes of models in the rest of the paper. *Boolean* models, *cost* models, and *probability* models. Abstracting away from the internal details for a moment, we can think of each of these models as defining a `score` function that assigns some value to a string:

$$\text{score} : \Sigma^* \rightarrow \mathcal{T}$$

where Σ is a set of symbols, Σ^* is the set of all possible strings generated from this set, and $\mathcal{T}$ is some set of values. The three models differ in what type of value the `score` function assigns.

2.1 Boolean models

We will use boolean models to correspond to the theoretical position that the phonotactic grammar is categorical, with gradience stemming from non-grammatical factors (Gorman, 2013; Kostyszyn and Heinz, 2022; Dai, 2025). The score function for these models assigns boolean values to strings:

$$\text{score} : \Sigma^* \rightarrow \{0, 1\}$$

Such models cannot represent a situation where the acceptability of /bnzk/ $\ll$ /bnık/ $\ll$ /blık/. If we take /bnzk/ to be ungrammatical and /blık/ to be grammatical, the model must place the intermediate form /bnık/ into one of these two categories.

2.2 Cost models

Cost models will correspond to the theoretical position that a gradient measure of acceptability is derived from a categorical grammar. There are many ways such a proposal could be implemented, but we will follow Durvasula (2020) and Kostyszyn and Heinz (2022), who derive such a gradient measure by counting the number of (categorical) constraints that a form violates. The `score` function for cost models assigns non-negative integer values to strings, with larger integers corresponding to lower phonotactic acceptability:

$$\text{score} : \Sigma^* \rightarrow \{0, 1, 2, \dots\}$$

In this model, acceptability is bounded on one side by 0, which corresponds to a "perfectly acceptable" form that violates no constraints. The other end of the scale is unbounded, since a form can violate arbitrarily many constraints. This means that, unlike the other two model types, we expect acceptability to *decrease* as the score increases. Such models can represent the case where the acceptability of /bnzk/ ≪ /bnık/ ≪ /blık/ by assigning the forms successively decreasing integer values.

2.3 Probability models

Probability models will correspond to the theoretical claim that gradience in acceptability corresponds directly to gradience in the grammar. Gradient grammars do not necessarily have to generate probabilities, but we will assume that is the case here. The score function for probability models is:

$$\text{score} : \Sigma^* \rightarrow [0, 1]$$

Such models can also represent the case where /bnzk/ ≪ /bnık/ ≪ /blık/ by assigning the forms successively increasing probabilities.

3 Turkish study

We will compare these three classes of models against new data from a large, online acceptability judgment study of Turkish nonce words.[2] This study expands on a previous acceptability judgment study on Turkish (Zimmer, 1969) by including a much larger number of stimuli and participants and using a slider task rather than a binary forced choice task. We will focus on *backness harmony* and *rounding harmony*, which are common in Turkic languages. Backness harmony requires vowels to agree in backness with the preceding vowel, while rounding harmony requires high vowels to agree in roundness with the preceding vowel (see Table 1). We can implement these restrictions using the following bigram constraints over vowel sequences:

- *[αback] [$-\alpha$back]: a vowel must agree in backness with the preceding vowel.
- *[αround] [$-\alpha$round, +high]: high vowels must agree in roundness with the preceding vowel.

These constraints govern suffix allomorphy: e.g., the plural form of /kedi/ 'cat' is [kedi-ler] 'cat-PL', while the plural of /kuʃ/ 'bird' is [kuʃ-lar] 'bird-PL'. Vowel harmony is is also evident as a strong tendency across the lexicon (though many disharmonic words exist, particularly loanwords) and in acceptability judgment tasks (Zimmer, 1969).

	[−back]		**[+back]**	
	[−round]	**[+round]**	**[−round]**	**[+round]**
[+high]	i	y	ɯ	u
[−high]	e	ø	a	o

Table 1: The vowel system of Turkish

3.1 Methodology

The stimuli consisted of 576 wug words with CVCVC shape. A Python script was used to generate every possible Turkish CVCVC word. Attested words found in the Turkish Electronic Living Lexicon (TELL; Inkelas et al., 2000) were automatically removed. Subsequent manual filtering was done by two native Turkish speakers. The remaining words were scored for unigram and Laplace-smoothed bigram probability using the UCI Phonotactic Calculator (Mayer et al., under revision) based on frequencies from citation forms in TELL. For each unique pair of vowels (8×8 total pairs), nine words were sampled such that they were distributed in a roughly uniform way across the unigram-bigram probability space. As a result, the mean probability of the tokens for each vowel pair was roughly the same (Fig. 1). The 576 tokens were synthesized to speech using Google Cloud. The recordings were vetted by the same two native Turkish speakers for naturalness and clarity.

The experiment was administered using Gorilla (`www.gorilla.sc` Anwyl-Irvine et al., 2020). All materials were presented in Turkish. After providing consent, participants completed a short demographic questionnaire. Participants then completed two screening tasks. The first was an audio check that asked them to identify a word presented to them acoustically. The second was a training run of the main experimental task, where participants were instructed to make a specific selection at the end as an attention check. Failure in either of these tasks led to exclusion from the experiment.

Finally, in the main experimental task, participants were asked to provide acceptability judgments of the stimuli based on their suitability as words in Turkish using a sliding, unnumbered scale. The right side of the scale corresponded to higher acceptability, and high-, mid-, and low-probability words were provided as landmarks (Fig. 2). Stimuli were presented with simultaneous audio and orthographic representation. Slider responses were represented on a numeric scale between 0 and 100, with 100 being the most acceptable.

[2] The data and code for this paper can be found at `https://github.com/connormayer/turkish_phonotactics`

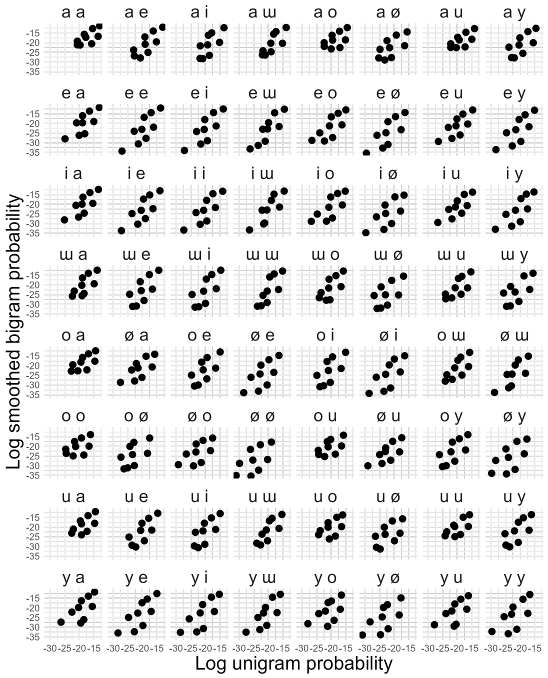


Figure 1: The distribution of unigram and bigram probabilities of the stimuli within each vowel group.

115 native speakers of Turkish were recruited using Prolific (www.prolific.com). 25 participants were excluded because they failed to provide consent or failed one of the two screening tasks. An additional 5 participants were excluded because they indicated in the demographic questionnaire that they had hearing impairment or that Turkish was not their native language. This left a total of 85 participants (38F; mostly age 25–35). Each participant rated 192 tokens after training and attention checks, leading to a total of 16,320 token ratings (about 28 ratings per word). Raw slider responses were normalized to z-scores within participant to control for idiosyncratic differences in mean and spread between participants.

3.2 Results

Fig. 3 shows participant responses broken down by harmonic class. Participants' responses reflect sensitivity to both backness and rounding harmony.

4 Modeling the Turkish data

In this section, we'll compare how well the different models described above predict the acceptability judgment data from the Turkish study. Crucially, each of these models employs the same set of possible constraints, differing only in the values they assign to each. This allows the effect of different value choices to be compared more directly.

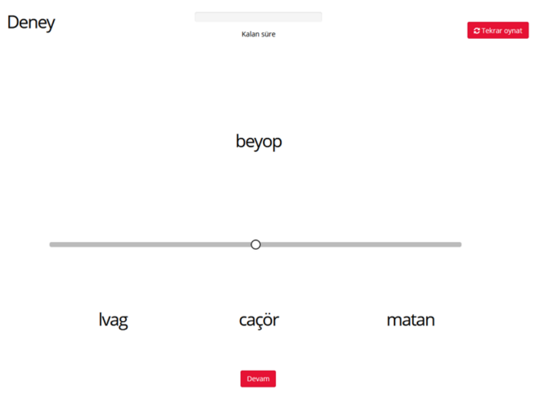


Figure 2: The experimental interface.

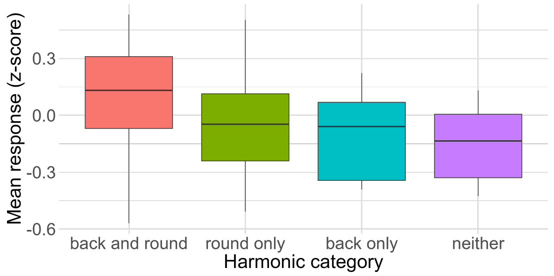


Figure 3: Normalized, mean participant responses broken down by harmonic category. Participants are sensitive to both backness and rounding harmony.

Because our interest is primarily in vowel harmony, we will use tier-based strictly local models with bigram constraints on the vowel tier (a TSL-2 model). It is beyond the scope of this paper to provide a full definition of TSL (see Heinz et al., 2011), but informally it means that we ignore consonants completely and assign scores based only on vowel bigrams. Bigrams can also reference word boundaries (#). This means the models are sensitive not only to which pairs of vowels occur in a word, but also which vowels begin and end the word.

Each model type has a Δ function that assigns a value to a bigram. These bigram values are then aggregated into the value returned by the `score` function discussed above.

4.1 Boolean models

Under a boolean model, the Δ function is:

$$\Delta_b : \Sigma^2 \rightarrow \{0, 1\}$$

where Σ^2 is the set of all possible bigrams, including the word boundary symbol . The boolean values assigned to each bigram in a string are aggregated into a single boolean by conjoining them:

$$\text{score}_b(x_1, \ldots, x_n) = \bigwedge_{i=1}^{n-1} \Delta_b(x_i, x_{i+1})$$

Legal and illegal bigrams receive scores of 1 and 0 respectively. The score for a string is 1 iff it contains only legal bigrams and 0 otherwise.

4.2 Cost models

Under a cost model, the Δ function is:

$$\Delta_c : \Sigma^2 \rightarrow \{0, 1, 2 \ldots\}$$

The integers assigned to each bigram are aggregated into a single integer score by summing them.

$$\text{score}_c(x_1, \ldots, x_n) = \sum_{i=1}^{n-1} \Delta_c(x_i, x_{i+1})$$

We will interpret the integer cost assigned to a bigram as the number of bigram constraints it violates. For example, a vowel bigram like /oi/ that violates both backness and rounding harmony might be assigned a cost of 2, while a bigram like /oy/ that violates only backness harmony might be assigned a cost of 1. Although these models could in principle represent varying constraint strengths by assigning different integer costs to each constraint, we will assume following previous work that all constraint violations are equally penalized (Durvasula, 2020; Kostyszyn and Heinz, 2022).

4.3 Probability model

Under a probability model, the Δ function is:

$$\Delta_p : \Sigma^2 \rightarrow [0, 1]$$

The probabilities for each bigram are aggregated into a single probability by taking their product:

$$\text{score}_p(x_1, \ldots, x_n) = \prod_{i=1}^{n-1} \Delta_p(x_i, x_{i+1})$$

The individual probabilities assigned to bigrams typically reflect their frequency (though this need not be the case). The probability assigned to a string reflects the probabilities of the bigram sequences it contains.

4.4 An example calculation

Consider again the vowel bigram /oi/. In Turkish, this may be dispreferred because it violates both backness and rounding harmony. Below I show how the score for this sequence can be calculated under each of the three types of models described above (we will discuss where the values assigned to each bigram come from in the following section).

$$\begin{aligned}\text{score}_b(\text{/oi/}) &= \Delta_b(\#\text{o}) \wedge \Delta_b(\text{oi}) \wedge \Delta_b(\text{i}\#) \\ &= 1 \wedge 0 \wedge 1 \\ &= 0\end{aligned}$$

$$\begin{aligned}\text{score}_c(\text{/oi/}) &= \Delta_c(\#\text{o}) + \Delta_c(\text{oi}) + \Delta_c(\text{i}\#) \\ &= 0 + 2 + 0 \\ &= 2\end{aligned}$$

$$\begin{aligned}\text{score}_p(\text{/oi/}) &= \Delta_p(\#\text{o}) \times \Delta_p(\text{oi}) \times \Delta_p(\text{i}\#) \\ &= 0.08 \times 0.107 \times 0.458 \\ &= 0.0004\end{aligned}$$

4.5 Defining Δ

A question that remains is how to actually define Δ for each model: that is, what specific values do we assign to each bigram? We will test several variants that differ in how Δ is defined.

4.6 Δ in the probability model

In the probability model, $\Delta_p(x, y)$ is defined to be $P(y|x)$, the conditional probability of the second sound in the bigram given the first. These probabilities were estimated using add-one smoothing (Chen and Goodman, 1999) from 18,472 citation forms in the TELL database (Inkelas et al., 2000) using the UCI Phonotactic Calculator (Mayer et al., in press). The conditional probabilities assigned to each bigram are shown in Fig. 4. Note that both backness harmony and rounding harmony are reflected in these probabilities: for the most part, harmonic sequences have higher probabilities than disharmonic ones (though other constraints are also apparent, such as a strong dispreference for /ø/ and /o/ in non-initial position).

The UCI Phonotactic Calculator returns log probabilities to avoid numerical underflow. The results in Section 4.7 use these log probabilities rather than the standard probabilities shown in Fig. 4.

4.6.1 Δ in the boolean model

We will test three variants of the boolean model. The first we will call the *harmony* model, based on

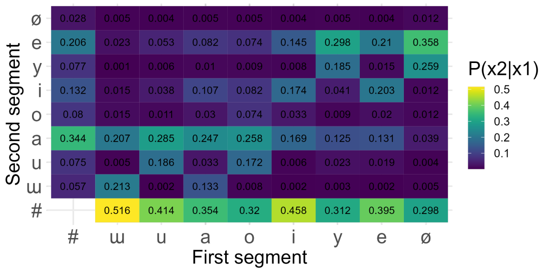


Figure 4: The probability model

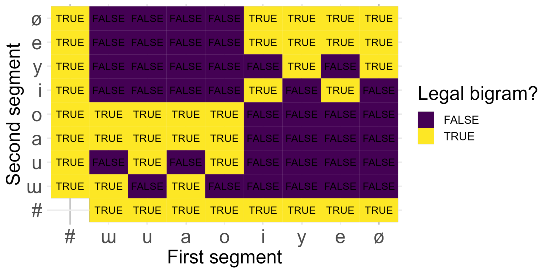


Figure 5: The boolean harmony model

Gorman (2013). Under this model, any bigram that violates either rounding or backness harmony (or both) receives a value of 0 and all other bigrams receive a value of 1. This model is shown in Fig. 5.

The second variant we will call the *exception filtering* model. This is a categorical Turkish phonotactic grammar from Dai (2025), which was learned by a statistical exception filtering process. For reasons of space I will not described the filtering process here, but it results in a more restrictive boolean model that still reflects backness and rounding harmony. This model is shown in Fig. 6.

The third variant we will call the *threshold* model. Under this model, a bigram is legal only if its conditional probability (as defined in the previous section) is above the 40th percentile of all the conditional bigram probabilities. The 40th percentile was opportunistically chosen because it maximized the performance of the model against this data. This is similar to the exception filtering model in that it is derived from frequencies in the lexicon, but it is generally more permissive. The values assigned by this model are shown in Fig. 7.

Gorman (2013) and Kostyszyn and Heinz (2022) also explore models where bigrams are only grammatical if they are attested. Unfortunately, all vowel bigrams are attested in TELL, which means such a model makes no predictions in this case.

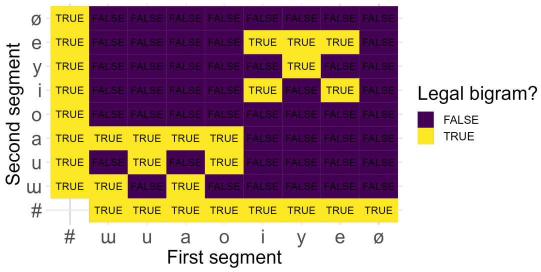


Figure 6: The boolean exception filtering model

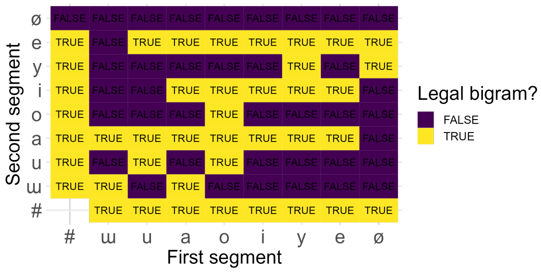


Figure 7: The boolean threshold model

4.6.2 Δ in the cost model

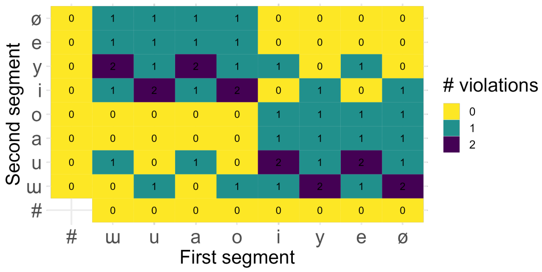
We consider only a single variant of the cost model, which uses the same bigram constraints as the harmony model but assigns them integer values instead. Bigrams that violate both backness and rounding harmony have a cost of 2; bigrams that violate one or the other have a cost of 1; and all other bigrams have a cost of 0. The values assigned to bigrams by this model are shown in Fig. 8.

4.7 Results

Each of the five models was used to score the 576 words from the acceptability judgment study. The model scores were correlated against the mean of the normalized acceptability scores for each word collected in the study. Table 2 reports Pearson, Kendall and Spearman correlations (See Albright, 2009, for some discussion of differences between these metrics in the context of phonotactics).

Value type	**Constraints**	r	τ	ρ
Probability	Cond. probs	**0.54**	0.36	**0.50**
Boolean	Threshold	0.46	**0.37**	0.45
Cost	Harmony	0.38	0.30	0.38
Boolean	Harmony	0.38	0.30	0.37
Boolean	Exception	0.36	0.27	0.33

Table 2: Correlations between model scores and mean acceptability judgments.



	#	ɯ	u	a	o	i	y	e	ø
ø	0	1	1	1	1	0	0	0	0
e	0	1	1	1	1	0	0	0	0
y	0	2	1	2	1	1	0	1	0
i	0	1	2	1	2	0	1	0	1
o	0	0	0	0	0	1	1	1	1
a	0	0	0	0	0	1	1	1	1
u	0	1	0	1	0	2	1	2	1
ɯ	0	0	1	0	1	1	2	1	2
#		0	0	0	0	0	0	0	0

Figure 8: The cost harmony model

These results generally support the probabilistic model as the best approximation of human acceptability judgments. The boolean threshold model comes the closest to matching its performance (and modestly surpasses it according to Kendall's τ). It is important to consider, however, that this model is derived from the conditional probability model: in other words, the best performing categorical model was produced by attending to gradience in the learning data. This is exactly the kind of model argued against by Chomsky (1957), where we "sharpen the blurred edges in the full statistical picture" (p. 17) by designating high probability forms as grammatical and low probability forms as ungrammatical.

Chomsky's objections aside, two natural questions the threshold model must deal with are (a) why the learner should track variability during acquisition only to discard it once the grammar is formed; and (b) how the threshold separating grammatical and ungrammatical structures is set. The learning algorithm in Dai (2025) uses a similar thresholding parameter to determine whether a bigram is exceptional or not. However, Dai finds that the best values of this threshold differ across data sets, and provides no principled way to derive it from the data. In contrast, the conditional bigram model is fit using maximum likelihood estimation, a robust and well-understood learning procedure.

These results favor the use of gradient models for modeling phonotactics. However, in the remainder of the paper I hope to convince you that the similarities between these models outweigh their differences.

5 Reconciling gradient and categorical models

Although these three model types differ in the values they assign to strings, there are many similarities in their basic structure. The boolean, cost, and probability models all assign some value to each segmental bigram (booleans, integers, or probabilities respectively) and aggregate them to get a single value for a string using some binary operation (conjunction, addition, or multiplication respectively). Approaching the models from this perspective, we can abstract away from the specific values and aggregation methods and express them in more mathematically general terms.

Δ maps bigrams to some set of values $\mathcal{T}$:

$$\Delta \colon \Sigma^2 \longrightarrow \mathcal{T}$$

Our `score` function aggregates these values using some binary operator $\owedge$ over $\mathcal{T}$:

$$\text{score}(x_1 \ldots x_n) = \bigowedge_{i=1}^{n-1} \Delta(x_i, x_{i+1})$$

The boolean, cost, and probability models described above can be instantiated from this more abstract model by specifying particular values of $\mathcal{T}$ and $\owedge$.

If $\owedge$ is associative and there is an identity element $\top$ in $\mathcal{T}$ such that $a \owedge \top = \top \owedge a = a$, which is the case for each of the set-operation pairs considered here, then $(\mathcal{T}, \owedge)$ forms a mathematical object called a *monoid*. Thinking in monoid-general terms allows us to take the same abstract model and parameterize it with different monoids. This means the same underlying model can compute different quantities, unifying models that appear to do vastly different things on the surface (Goodman, 1999; Eisner, 2003; Chandlee and Heinz, 2017). In other words, we can separate the structure of the model from the values it computes.

In addition to the monoids discussed above, our humble bigram model can actually compute a range of other useful quantities, such as constraint violation profiles using the monoid $(\mathbf{N}^k, +)$, where $\mathbf{N}^k$ is the set of vectors of natural numbers of length k (e.g. Riggle, 2009), or even input SL-2 string transduction (e.g. Chandlee, 2014) using the monoid $(\Sigma^*, \cdot)$, where $\cdot$ is a string concatenation operator.

Most of the models we work with in formal language theory, such as subregular models (Heinz, 2018), finite-state automata, context-free grammars, and so on, can be expressed in these general terms. Although non-deterministic models require an additional operator to combine multiple parses of the same string, a more complex mathematical structure called a *semiring* can be used analogously to monoids for such models.[3]

[3] The probability monoid/semiring is usually defined to

5.1 Monoids in phonology

Why is the idea of monoids useful for us as phonologists? An example comes from the domain of semantics: Giorgolo and Asudeh (2014) apply different semirings to the same underlying semantic model to capture differences between heuristic and mathematical reasoning. They suggest that the underlying structure of both reasoning processes is the same, but that these processes can generate different types of outcomes depending on the context (in this case, how important it is to be precise).

There's perhaps an analogy to be made here with our categorical and gradient models of Turkish. It is clear from the results above and past work on Turkish that vowel harmony is centrally important for both suffix allomorphy and phonotactics (it is striking how much of the variation in participants' responses above can be captured by only attending to the vowels in each word). However, these sensitivities manifest in different ways in each domain. Harmony constraints are essentially categorical when determining suffix allomorphy (it's always [kediler] and never *[kedi-lar]), but these constraints provide only a gradient preference when determining word acceptability.

Even if we choose to treat alternations as essentially categorical and phonotactics as essentially gradient, our categorical and gradient models have more in common than might be evident at first glance. Each of the models we discussed in this paper are TSL-2 grammars: they employ the same types of representations (segments, constraints, etc.); they operate on the vowel tier; they are sensitive only to constraints between adjacent vowels; and they disprefer the same types of structures. The fact that these same representations and dependencies appear to be necessary for modeling both gradient and categorical phenomena suggest that both are governed at least in part by the same underlying linguistic system (Hayes, 2000), and past work has claimed that there is a close connection between the acqusition of alternations and phonotactics (e.g. Hayes, 2004; Chong, 2021; Jun et al., 2025)

6 Conclusion

Durvasula (2020) implores us to prioritize categorical models of phonotactics so that we can "focus on what's a possible constraint or rule" and "commit to a specific set of representations." I contend that this is a false dichotomy: constraints and representations in the grammar can be studied independently of the values the grammar assigns. This flexibility allows us to engage with a broader range of empirical phenomena for which categorical or gradient models provide better approximations while still relating these phenomena to the same core linguistic knowledge (Hayes, 2000). Although the results of this study support the position that phonotactic knowledge is best captured using gradient models, we can gain insight into the representations and dependencies in the linguistic grammar by considering both types of models.

Acknowledgments

Thanks to Huteng Dai, Karthik Durvasula, Richard Futrell, Kyle Gorman, Jeff Heinz, Jon Rawski, Megha Sundara, Bernard Tranel, three anonymous reviewers, and the attendees of the 2024 Workshop on Formal Language Theory in Morphology and Phonology at the 2024 Meeting of the Linguistic Society of America for their useful feedback and discussions. Thanks also to Cem Babalik and Defne Bilhan for their assistance in designing the stimuli for the Turkish study. This work was supported by NSF Award #2214017.

References

Adam Albright. 2009. Feature-based generalisation as a source of gradient acceptability. *Phonology*, 26(1):9–41.

Alexander L Anwyl-Irvine, Jessica Massonnié, Adam Flitton, Natasha Kirkham, and Jo K Evershed. 2020. Gorilla in our midst: An online behavioral experiment builder. *Behavior research methods*, 52:388–407.

Todd M Bailey and Ulrike Hahn. 2001. Determinants of wordlikeness: Phonotactics or lexical neighborhoods? *Journal of Memory and Language*, 44(4):568–591.

Jane Chandlee. 2014. *Strictly local phonological processes*. University of Delaware.

Jane Chandlee and Jeffrey Heinz. 2017. Computational phonology. In *Oxford Research Encyclopedia of Linguistics*.

assign values from $\mathcal{R}$, with the additional implicit restriction that the assigned values must form a valid probability distribution. There are non-trivial issues that arise in choosing exactly *which* particular values (or *weights*, to use the more technical term) our model should assign, such as normalization in probabilistic models, whether the order of the values is total and monotonic, etc. These considerations are not the focus of this paper.

Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393.

Eleanor Chodroff and Colin Wilson. 2014. Phonetic vs. phonological factors in coronal-to-dorsal perceptual assimilation. Paper presented at LabPhon 14: the 14th Conference on Laboratory Phonology, Tokyo.

Noam Chomsky. 1957. *Syntactic structures*. Walter de Gruyter.

Noam Chomsky and Morris Halle. 1968. *The sound pattern of English*. Harper & Row, New York.

Adam J Chong. 2021. The effect of phonotactics on alternation learning. *Language*, 97(2):213–244.

John Coleman and Janet Pierrehumbert. 1997. Stochastic phonological grammars and acceptability. In John Coleman, editor, *Proceedings of the 3rd Meeting of the ACL Special Interest Group in Computational Phonology*, pages 49–56. Association for Computational Linguistics, Somerset, NJ.

Huteng Dai. 2025. An exception-filtering approach to phonotactic learning. *Phonology*, 42:e5.

Huteng Dai, Connor Mayer, and Richard Futrell. 2023. Rethinking representations: A log-bilinear model of phonotactics. *Proceedings of the Society for Computation in Linguistics*, 6.

Robert Daland, Bruce Hayes, James White, Marc Garellek, Andreas Davis, and Ingrid Normann. 2011. Explaining sonority projection effects. *Phonology*, 28:197–234.

Emmanuel Dupoux, Erika Parlato, Sonia Frota, Yuki Hirose, and Sharon Peperkamp. 2011. Where do illusory vowels come from? *Journal of memory and language*, 64(3):199–210.

Karthik Durvasula. 2020. O gradience, whence do you come? Keynote presentation at the 2020 Annual Meeting on Phonology.

Jan Edwards, Mary E Beckman, and Benjamin Munson. 2004. The interaction between vocabulary size and phonotactic probability effects on children's production accuracy and fluency in nonword repetition.

Jason Eisner. 2003. Simpler and more general minimization for weighted finite-state automata. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 64–71.

G. Giorgolo and A. Asudeh. 2014. One semiring to rule them all. In *Proceedings of the 36th Annual Meeting of the Cognitive Science Society*, pages 208–226. Cognitive Science Society, Québec City.

Matthew Goldrick and Meredith Larson. 2008. Phonotactic probability influences speech production. *Cognition*, 107(3):1155–1164.

Sharon Goldwater and Mark Johnson. 2003. Learning OT constraint rankings using a maximum entropy model. In Jennifer Spenader, Anders Eriksson, and Östen Dahl, editors, *Proceedings of the Stockholm Workshop on Variation within Optimality Theory*, pages 111–120. Stockholm University, Department of Linguistics, Stockholm.

Josh Goodman. 1999. Semiring parsing. *Computational Linguistics*, 25(4):573–605.

Kyle Gorman. 2013. *Generative phonotactics*. Ph.D. thesis, University of Pennsylvania.

Bruce Hayes. 2000. *Gradient well-formedness in Optimality Theory*, pages 88–120. Oxford University Press.

Bruce Hayes. 2004. Phonological acquisition in optimality theory: The early stages. *Constraints in Phonological Acquisition/Cambridge University Press*.

Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic inquiry*, 39(3):379–440.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. *Phonological typology, phonetics and phonology*, 23:126–195.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints in phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64.

Sharon Inkelas, Aylin Küntay, Orhan Orgun, and Ronald Sprouse. 2000. Turkish electronic living lexicon (TELL). *Turkic Languages*, 4:253–275.

Jongho Jun, Hanyoung Byun, Seon Park, and Yoona Yee. 2025. How tight is the link between alternations and phonotactics? *Phonology*, 42:e3.

Jimin Kahng and Karthik Durvasula. 2023. Can you judge what you don't hear? Perception as a source of gradient wordlikeness judgments. *Glossa*, 8(1).

Kalina Kostyszyn and Jeffrey Heinz. 2022. Categorical account of gradient acceptability of word-initial Polish onsets. In *Proceedings of AMP 2021*.

Andrey Andreyevich Markov. 1913. An example of statistical investigation in the text of 'Eugene Onyegin' illustrating coupling of 'tests' in chains. *Proceedings of the Academy of Sciences of St. Petersburg*, 7:153–162.

Sven L Mattys, Peter W Jusczyk, Paul A Luce, and James L Morgan. 1999. Phonotactic and prosodic effects on word segmentation in infants. *Cognitive Psychology*, 38(4):465–494.

Connor Mayer. 2021. Capturing gradience in long-distance phonology using probabilistic tier-based strictly local grammars. In *Proceedings of the Society for Computation in Linguistics 2021*, pages 39–50, Online. Association for Computational Linguistics.

Connor Mayer, Arya Kondur, and Megha Sundara. in press. The UCI Phonotactic Calculator: An online tool for computing phonotactic metrics. *Behavior Research Methods*.

Connor Mayer, Arya Kondur, and Megha Sundara. under revision. The UCI Phonotactic Calculator: An online tool for computing phonotactic metrics. *Behavior Research Methods*.

Connor Mayer, Adeline Tan, and Kie Ross Zuraw. 2024. Introducing maxent. ot: an R package for Maximum Entropy constraint grammars. *Phonological Data and Analysis*, 6(4):1–44.

James M McQueen. 1998. Segmentation of continuous speech using phonotactics. *Journal of Memory and Language*, 39(1):21–46.

Karima Mersad and Thierry Nazzi. 2011. Transitional probabilities and positional frequency phonotactics in a hierarchical model of speech segmentation. *Memory & Cognition*, 39:1085–1093.

Dennis Norris and James M. McQueen. 2008. Shortlist B: a Bayesian model of continuous speech recognition. *Psychological Review*, 115:357–395.

Alan Prince and Paul Smolensky. 1993/2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell.

Jason Riggle. 2009. Violation semirings in optimality theory. *Research on Language and Computation*, 7:1–12.

Robert Scholes. 1966. *Phonotactic grammaticality*. Mouton, The Hague.

Carson Schütze. 1996. *The empirical base of linguistics: Grammaticality judgments and linguistic methodology*. The University of Chicago.

Claude E. Shannon. 1948. A mathematical theory of communication. *Bell System Technical Journal*, 27:623–656.

Jeremy Steffman and Megha Sundara. 2023. Short-term exposure alters adult listeners' perception of segmental phonotactics. *JASA Express Letters*, 3(12).

Jeremy Steffman and Megha Sundara. 2024. Disentangling the role of biphone probability from neighborhood density in the perception of nonwords. *Language and Speech*, 67(1):166–202.

Holly L Storkel. 2001. Learning new words. *Journal of Speech, Language, and Hearing Research*, 44(6):1321–1337.

Conrad F Taylor and George Houghton. 2005. Learning artificial phonotactic constraints: time course, durability, and relationship to natural constraints. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 31(6):1398.

Michael S Vitevitch and Paul A Luce. 1999. Probabilistic phonotactics and neighborhood activation in spoken word recognition. *Journal of memory and language*, 40(3):374–408.

Jill A Warker. 2013. Investigating the retention and time course of phonotactic constraint learning from production experience. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 39(1):96.

Jill A Warker and Gary S Dell. 2006. Speech errors reflect newly learned phonotactic constraints. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 32(2):387.

Jill A Warker and Gary S Dell. 2015. New phonotactic constraints learned implicitly by producing syllable strings generalize to the production of new syllables. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 41(6):1902.

K.E. Zimmer. 1969. Psychological correlates of some Turkish morpheme structure conditions. *Language*, pages 309–321.

Learning Covert URs via Disparity Minimization

Jonathan Charles Paramore
University of California, Santa Cruz
jcparamo@ucsc.edu

Abstract

When considering the acquisition of underlying representations (URs), two common challenges are often levied against the inclusion of abstract URs in phonological theory: (1) permitting abstract URs causes the search space of potential URs to grow to a computationally intractable degree, and (2) learners have no recourse through which to prefer minimally abstract URs over increasingly abstract alternatives when both types of URs model the data with equal success. This paper directly addresses the second issue by implementing a MaxEnt learner equipped with a bias that penalizes disparities between UR inputs and their corresponding outputs. By favoring mappings with minimal divergence, the bias generates a preference for minimally abstract URs when competing candidates perform equally well in modeling the data. In addition, the paper proposes a conceptual framework for addressing the first issue, in which the space of potential URs is organized so that candidates are considered serially, beginning with those that exhibit the fewest disparities. This method offers a potential strategy for avoiding the added compute time introduced by permitting UR abstraction.

1 Introduction

A subject of significant debate since the advent of generative phonology concerns the level of abstraction that underlying representations (URs) are permitted to assume (Kenstowicz and Kisseberth, 1979). Classic generative phonology holds the rather strong position that a UR can be completely *covert* in relation to all of its allomorphs, never showing its true identity in surface forms. However, from a learning perspective, permitting this level of abstraction poses serious challenges. One of the most compelling objections is that covert URs render the learning problem intractable. Two key difficulties arise. First, the space of potential URs that a learner must consider becomes prohibitively large. When highly abstract URs are allowed, the search space expands dramatically, exceeding what can feasibly be explored in its entirety by a learner (Albright, 2002; Jarosz, 2015, 2019; Wang and Hayes, 2025).

Most models attempt to solve this issue by curtailing the level of abstraction URs can take, in essence shrinking the search space to a manageable size. For instance, Wang and Hayes (2025) constrain the search space by restricting the abstractness of candidate URs using a hierarchy of representational abstraction defined in Kenstowicz and Kisseberth (1977, ch.1). The model is impressive and successfully accounts for analyses at various levels of abstraction, but it fails to account for datasets requiring covert URs, like the Punjabi nasality pattern considered in this paper.

The second issue that arises when learning covert URs is that the learner has no means through which to prefer a less abstract UR over a highly abstract UR if both representations succeed in modeling the data. One particularly promising approach aimed at alleviating this computational burden is outlined in O'Hara (2017) with the use of a Maximum Entropy (MaxEnt) grammar called MaxLex. O'Hara provides compelling evidence from Klamath showing that a stem-final [i]-[ø] alternation in words like [ʔeːw-a] 'is deep' ∼ [ʔeːwitk^h] 'deep' cannot be captured by either epenthesis or deletion but instead requires a covert UR, /e/, that deletes when not in the initial syllable, unless deletion would produce an illicit consonant cluster, in which case /e/ is raised to [i]. Importantly, /e/ is covert in the stem-final position of stems like /ʔeːwe/ because it never surfaces in any allomorph. Moreover, O'Hara demonstrates that MaxLex has an *emergent* preference for minimally abstract URs, driven by an L2 Gaussian Prior that attempts to minimize increases in the weights of faithfulness constraints.

In this paper, I primarily address how the learner might come to prefer minimal UR abstraction. I first show that MaxLex *fails* to prefer minimally abstract URs over increasingly abstract alternatives for a set of non-alternating pre-nasal vowels in Pakistani Punjabi (Paramore, 2023). This failure arises because both the minimally abstract UR and more abstract alternatives provide equally accurate accounts of the data and require identical changes in faithfulness constraint weights to do so. As a solution, I propose an updated MaxLex learner equipped with a disparity bias that penalizes changes in UR→SR mappings. The effect of this bias is that, if two URs model a set of data equally well and do not differ in the minimization of the MaxLex L2 prior, the learner selects the UR that generates the minimum number of disparities. In addition to creating a preference for minimal UR abstraction, this disparity bias has potential to provide a mechanism through which the learner can efficiently search the space of potential URs without needing to stipulate its contents, as discussed in section 6.

2 MaxLex

The basic learning procedure taken by MaxLex is similar to other MaxEnt learning models (e.g. Hayes and Wilson, 2008; Pater et al., 2012; Wang and Hayes, 2025). Two general stages characterize the process. In the first stage, the learner is oblivious to morphological alternations and paradigmatic relations, and, as a consequence, the identity of underlying forms and mappings from those underlying forms to surface realizations is not considered. Instead, the learner has been confronted with a wealth of linguistic data and focuses on acquiring fluency in language-specific phonotactics, an aspect of the grammar that remains unchanged regardless of what the underlying forms turn out to be.

In computational terms, at the outset of the phonotactic stage, MaxLex is fed a batch of data, a set of constraints with intermediate weights (e.g., 50), and the parameters for what constitutes a violation. Equipped with this information, the learner uses gradient descent optimization to minimize an objective function (in this case, the negative log-likelihood of the data) by adjusting the constraint weights appropriately until it arrives at the minimum possible value. A grammar with a 100% probability of producing the observed data will have an objective function value of zero, but a grammar with only a 50% probability of producing the observed data will result in a much higher objective function value.

In the second stage of learning, MaxLex becomes morphologically aware, understanding that words are constructed from morphemes, and those morphemes sometimes appear in phonologically distinct ways, depending on the context. For instance, during the phonotactic stage, the learner ignores the morphological relationship between the Punjabi words [sɑɑ] 'breath' and [sɑ̃ɑ̃ʊ̃ɑ̃ɑ̃] 'breaths', focusing only on phonotactic well-formedness. In the morphologically aware stage, however, the learner has discovered that the same morpheme for 'breath' occurs in both words and seeks to assign a single UR that can map to both of the observed forms. As such, the learner is confronted with a more complex learning problem in which it must work to determine what combination of constraint weights and underlying form probabilities maximizes the likelihood of observing the data to which it has been exposed (Jarosz, 2006a,b).

A crucial aspect of the morphologically aware learning stage that MaxLex capitalizes on is the way in which abstraction is mitigated in the choice of potential URs. Specifically, the objective function in MaxLex is constructed from the negative log-likelihood of the data plus the value of an L2 Gaussian Prior that prefers to use constraints active elsewhere in the grammar to account for abstract phonological patterns rather than altering the weight of novel constraints to accomplish the same task.[1] The negative log-

[1] Both Pater et al. (2012, p.66) and Wang and Hayes

likelihood (NLL) of a dataset, given in equation 1, is calculated by determining the combination of constraint weight (**w**) and UR probability (***π***) values that maximize the likelihood (thereby minimizing the NLL) of observing a set of observed words (O$_i$ - O$_n$).

$$NLL = -\ln \left[\prod_{i=1}^{n} \left(\mathbb{P}[O_i \mid (\mathbf{w}, \boldsymbol{\pi})] \right) \right] \qquad (1)$$

To increase grammar restrictivity, the L2 Gaussian prior shown in equation 2 inherently favors markedness constraints with maximum weights of 100 and faithfulness constraints with minimum weights of zero. This bias is implemented by taking the squared difference of actual weight values (w_i) from their ideal weight (c_i).[2] If, however, the language data confronted by the learner indicates that different constraint weights would improve the success of the grammar in modeling the data (i.e., sufficiently lowering the NLL), these biases can be overcome. Thus, if a faithfulness constraint is given a non-zero weight to model some phonotactic pattern in the first stage of language learning, that same constraint will be preferred over a novel constraint with a zero weight to model another pattern concerning underlying forms, assuming both constraints can account for the observed data equally well. This preference to use the already-active faithfulness constraint falls out from the fact that the MaxLex prior seeks to minimize deviations in constraint weights from their optimal values. Because of this, O'Hara argues that a segment's UR is naturally restricted in its potential for abstraction by this bias.

$$\mathcal{O}_{Lex}(\mathbf{w}, \boldsymbol{\pi}) = \boldsymbol{NLL} + \underbrace{\sum_{w_i \in \mathbf{w}} \frac{(w_i - c_i)^2}{\sigma_i^2}}_{\text{L2 Gaussian Prior}} \qquad (2)$$

The success of MaxLex in learning covert URs is demonstrated by examining a stem-final [i]∼[ø] alternation in a set of Klamath verbs, which, as O'Hara (2017) shows, capitalizes on a faithfulness constraint that is active in another area of the grammar to account for the alternation. As O'Hara delineates in detail in his computational proof, Maxlex takes advantage of these faithfulness constraint weight differences when deciding upon the optimal covert UR. However, that same learning process used to constrain UR abstraction in the Klamath [i]∼[ø] alternation is unavailable for the URs of non-alternating pre-N vowels in Punjabi.

3 Pakistani Punjabi

Pakistani Punjabi is an Indo-Aryan language spoken by about 78 million people, primarily in the Punjab province of Pakistan (Bashir and Conners, 2019). Long vowels in Punjabi contrast in nasality, but this contrast is neutralized before nasal consonants (e.g., [tɑɑ] 'warmth' vs. [tɑ̃ɑ̃] 'that' but [tɑ̃ɑ̃n] 'melody' vs. *[tɑɑn]). Additionally, Punjabi exhibits a process of nasal harmony, in which contrastive /ṼṼ/ vowels trigger the leftward spread of nasalization, with glides and vowels participating and other consonants acting as blockers, as shown in Table 1i. Pre-N vowels, on the other hand, surface as categorically nasalized and phonetically identical to contrastive /ṼṼ/ vowels, but they do not trigger nasal harmony (Table 1ii) (Paramore, 2023).

To account for the phonetic indistinguishability of /VVN/ and contrastive /ṼṼ/ vowels in terms of their nasality coupled with the fact that only contrastive /ṼṼ/ vowels trigger nasal harmony in Punjabi, /VVN/ vowels must be analyzed as underlyingly [-nas] without ever surfacing as such. In this view, the nasal harmony pattern in Punjabi serves as a straightforward example of counterfeeding opacity, in which underlyingly oral pre-N vowels undergo a predictable process of nasalization. Nevertheless, only underlying /ṼṼ/ vowels trigger nasal harmony. Harmony in Punjabi is thus sensitive to whether a vowel is underlyingly oral or nasal – even for vowels that are always *phonetically* nasal. This implies that /VVN/ vowels have abstract oral URs that are consistently distinct from their phonetic forms.

(2025, p.17, 34-35) incorporate similar biases favoring markedness constraints over faithfulness constraints.

[2] c_i is set to 100 for markedness constraints and zero for faithfulness constraints. O'Hara (2017) uses σ_i^2 as a plasticity constant (which he sets at 20 for markedness constraints and 25 for faithfulness constraints) to modulate how much deviations from ideal weights impact the value of the objective function.

i.	/sɑɑ-ʋɑ̃ɑ̃/	→	[sɑ̃ɑ̃ʋ̃ɑ̃ɑ̃]	'breath-PL'
ii.	/tɑɑʋɑɑn/	→	[tɑɑʋɑ̃ɑ̃n]	'penalty'

Table 1: Nasal Harmony in Punjabi.

i.	[sɑɑ]	'breath'	ii.	[sɑ̃ɑ̃ʋ̃ɑ̃ɑ̃]	'breaths'
iii.	[ʊʃɑɑ]	'morning'	iv.	[ʊʃɑ̃ɑ̃ʋ̃ɑ̃ɑ̃]	'mornings'
v.	[gɑ̃ɑ̃]	'cow'	vi.	[gɑ̃ɑ̃ʋ̃ɑ̃ɑ̃]	'cows'
vii.	[tʃʰɑ̃ɑ̃]	'shade'	viii.	[tʃʰɑ̃ɑ̃ʋ̃ɑ̃ɑ̃]	'shades'
ix.	[tɑɑʋɑ̃ɑ̃n]	'penalty'	x.	[prəʋɑ̃ɑ̃n]	'accepted'

Table 2: Punjabi surface forms fed to MaxLex

4 MaxLex and Punjabi pre-N vowels

In attempting to learn the opaque nasalization patterns in Punjabi, MaxLex begins with an initial phonotactic learning stage. The observed data fed to the learner is given in Table 2. Forms 2i-iv show that underlyingly oral vowels are nasalized via nasal harmony when the appropriate suffix is attached (in this case, the plural marker). The forms in 2v-viii show the learner that a nasality contrast exists for vowels; otherwise, the learner may choose to analyze the vowels in 2i-ii as underlyingly nasal to explain the nasal harmony distinctions found between /VVN/ and contrastive /ṼṼ/ vowels. Finally, the forms in 2ix-x provide the learner with examples of the underapplication of nasal harmony for non-alternating /VVN/ vowels.

Individual Python scripts were developed for the phonotactic learning stage and morphologically aware learning stage to carry out the computational optimizations. The constraints used in the learner are provided in Table 3 with the initial weights set at 50, along with the weights acquired in the phonotactic learning stage in the rightmost column. Most of these constraints are straightforward, but a few merit further explanation.[3] First, as is well known, the standard parallel evaluation architecture of MaxEnt learners presents difficulty for the successful acquisition of opaque processes like nasal harmony in Punjabi (McCarthy, 2000, 2007). To handle this, I choose to analyze the nasality patterns using contextual faithfulness constraints (Hauser and Hughto, 2020), but other approaches capable of handling counterfeeding opacity in a parallel framework are equally viable. At its root, the contextual faithfulness constraint schema penalizes changes to a specified feature for a segment that occurs in a specified context in the input. The contextual faithfulness constraint relevant to the Punjabi nasalization data, Id[nas]/_V, penalizes changes in nasality to a segment occurring before a vowel that is oral in the input. When high-ranked, this constraint precludes underlying oral vowels – as /VVN/ vowels are proposed to be here – from continuing the transmission of nasal harmony to its immediately preceding segment.

Another important note is the inclusion of Id[rd] and *LowRd in the constraint set. For reasons that will become clearer when discussing the updated learning algorithm in section 5, I provide the learner with two potential covert URs to choose between. The restrictedly abstract and intuitively most appealing covert UR for a /VVN/ vowel like [ɑ̃ɑ̃] in [tɑɑʋɑ̃ɑ̃n] is /ɑɑ/. /ɑɑ/ possesses an identical feature set to [ɑ̃ɑ̃] except for one disparity: nasality. Because nasality is the key underlying feature that results in distinct harmony patterns for /VVN/ and contrastive /ṼṼ/ vowels, it makes sense for nasality to be the only feature that changes between the UR and SR of /VVN/ vowels. With that said, MaxLex does not contain an inherent mechanism to act upon this sensible conclusion. Instead, the learner is free to choose any covert UR that models the data and minimizes changes in constraint weights from their biases, regardless of whether there are one or fifty feature disparities in the UR→SR mapping.

To focus on the learner's preference for minimally abstract URs, I provide MaxLex with one additional potential covert UR, /ɒɒ/. Just like its unrounded counterpart /ɑɑ/, the low round back vowel /ɒɒ/ is quite similar to its corresponding SR, [ɑ̃ɑ̃], except it contains *two* disparities rather than one: nasality and roundedness. Importantly, any increasingly abstract UR (e.g., diacritics) would suffice in the following discussion, but /ɒɒ/ is an especially good candidate because it is *more* abstract than /ɑɑ/ (/ɒɒ/ never surfaces in Punjabi and has more disparities in the input-output mapping) but only minimally so. Thus, /ɒɒ/ serves as a stand-in for any overly abstract covert UR that needs to be ruled out,

[3] See 5 in the appendix for a full set of constraint definitions.

Constraint	Type	initial w	final w
ID[nas]	faith.	50.00	51.37
IDFIN[nas]	faith.	50.00	44.83
SPRD-L[nas]	mark.	50.00	92.83
*NASOBS	mark.	50.00	100.00
*NASG	mark.	50.00	99.48
ID[nas]/_V	contfaith.	50.00	100.00
*VVN	mark.	50.00	100.00
ID[rd]	faith.	50.00	0.00
*LOWRD	mark.	50.00	100.00

Table 3: Constraint weights after phonotactic learning with MaxLex.

and if /ɒɒ/ is ruled out, potential URs with greater disparities will also be ruled out.[4]

The weights acquired in the phonotactic learning stage of MaxLex demonstrate three phonotactic restrictions in Punjabi that must hold regardless of the particular UR chosen for /VVN/ vowels. First, low round vowels never surface in Punjabi, so *LOWRD is undominated and ID[rd] is inactive and set to zero. As shown in (1), this weighting relationship appropriately unrounds all inputs containing /ɒɒ/ with a probability of 1.0.

(1) Low Round vowels never surface

/sɒɒ/	*LOWRD 100.00	ID[rd] 0.00	$\mathcal{H}$	$\tilde{\mathcal{P}}$
a. ☞ sɑɑ		-1	0	1.0
b. sɒɒ	-1		-100	$4e^{-44}$

Another phonotactic restriction MaxLex acquires is the absolute ban on nasal obstruents in Punjabi. To accomplish this, *NASOBS must outweigh SPRD-L, as in (2).

(2) Obstruents never nasalized

/sɑɑʋɑ̃ɑ̃/	*NASOBS 100.00	SPRD-L 92.83	$\mathcal{H}$	$\tilde{\mathcal{P}}$
a. ☞ sɑ̃ɑ̃ʋ̃ɑ̃ɑ̃		-1	-92.83	0.999
b. s̃ɑ̃ɑ̃ʋ̃ɑ̃ɑ̃	-1		-100	$8e^{-4}$

Finally, in order for /VVN/ vowels to surface consistently as nasal vowels, either *VVN or SPRD-L must outweigh ID[nas]. In fact, both constraints end up outweighing ID[nas], resulting in /VVN/ vowels always surfacing as nasal, as in (3).

Constraint	Type	initial w	final w
ID[nas]	faith.	51.37	3.36
IDFIN[nas]	faith.	44.83	99.96
SPRD-L	mark.	92.83	5.65
*NASOBS	mark.	100.00	100.00
*NASG	mark.	99.48	0.19
ID[nas]/_V	contfaith.	100.00	100.00
*VVN	mark.	100.00	100.00
ID[rd]	faith.	0.00	0.00
*LOWRD	mark.	100.00	100.00

UR	$\mathcal{P}$
/tɑɑʋɑ̃ɑ̃n/	1.0

Table 4: Constraint weights and UR probabilities with concrete URs only

(3) /VVN/ vowels always nasalized

/siin/	*VVN 100.00	ID[nas] 51.37	SPRD-L 92.83	$\mathcal{H}$	$\tilde{\mathcal{P}}$
a. ☞ sĩĩn		-1	-1	-144.2	1.0
b. siin	-1		-2	-285.66	$3e^{-62}$

Once the morphologically aware learning stage begins, MaxLex recognizes that surface alternations such as [sɑɑ] and [sɑ̃ɑ̃] belong to the same underlying morpheme. We will first consider the use of concrete URs to model the data. For our purposes, the important morphemes are those containing non-alternating pre-N vowels like [tɑɑʋɑ̃ɑ̃n]. Because [tɑɑʋɑ̃ɑ̃n] only exhibits a single surface form, only one concrete UR is available to MaxLex, and using it prevents MaxLex from accurately modeling the data. The results for constraint weights and UR probabilities with only concrete URs are given in Table 4. Again, because [tɑɑʋɑ̃ɑ̃n] does not exhibit morphological alternations, there is only one potential UR, and it receives all of the probability as the correct UR for modeling the data.

However, using only concrete URs results in the model's inability to successfully learn the appropriate constraint weights and an almost zero probability of learning the correct nasalization pattern of forms with /VVN/ vowels. This is exemplified by the tableau in (4). Because the URs for both /VVN/ and contrastive /ṼṼ/ vowels are identical, MaxLex cannot correctly learn the pattern. When presented with /tɑɑʋɑ̃ɑ̃n/, the learner incorrectly as-

[4] Note that a covert UR like the *nasalized low back round vowel /ɒ̃ɒ̃/* only has a single disparity in its mapping to [ɑ̃ɑ̃] (roundedness), so it would tie /ɑɑ/ in its performance on the disparity component of the objective function. However, just like the concrete UR /ɑ̃ɑ̃/ fails to model the lack of harmony triggered by /VVN/ vowels in Punjabi, any other nasal vowel would run into the same issue.

signs almost all the probability to the candidate that exhibits nasal harmony.

(4) Failure of Concrete URs to model Punjabi nasalization

/tɑɑʋɑ̃ɑ̃n/	*VVN 100.00	SPRD-L 5.65	*NASG 0.19	ID[nas] 3.36	$\mathcal{H}$	$\tilde{\mathcal{P}}$
a. tɑɑʋɑ̃ɑ̃n		-3			-16.95	0.012
b. tɑɑʋɑɑn	-1	-4		-1	-125.96	$6e^{-50}$
c. tɑ̃ɑ̃ʋ̃ɑ̃ɑ̃n		-1	-1	-2	-12.56	0.988

Up to this point, the learning process has followed the same general pattern as the Klamath [i]-[ø] alternation discussed in O'Hara (2017). The phonotactic patterns were learned, and using a concrete UR for /VVN/ vowels resulted in a failure to accurately predict the observed data. Now, just as for Klamath, MaxLex is provided two covert URs to consider when modeling the data. The results of the morphologically aware learning stage with /ɑ̃ɑ̃/, /ɑɑ/, and /ɒɒ/ included as potential URs are provided in Table 5. Here, the final constraint weights are quite similar to the weights when concrete URs were the only potential option, but the inclusion of the covert representations as potential URs for forms with /VVN/ vowels allows MaxLex to accurately model the data, with a .98 total probability of observing the correct surface forms for all words fed to the learner. However, while MaxLex is successful in modeling the data with the inclusion of these two covert URs, it is unsuccessful in discriminating between them, instead assigning an equal 0.5 probability to both covert URs. In other words, the MaxLex prior cannot distinguish between a restrictedly abstract UR like /ɑɑ/ and an unnecessarily abstract UR like /ɒɒ/. The reason for this is that changes in constraint weights from the phonotactic to the morphologically-aware learning stage are identical regardless of which covert UR is used. To permit the nasal harmony pattern in forms with contrastive /ṼṼ/ vowels, ID[nas] and *NASG need to lower so that their combined sum is less than SPRD-L. This change holds regardless of whether the UR for the /VVN/ vowel in [tɑɑʋɑ̃ɑ̃n] is /ɑɑ/ or /ɒɒ/. Additionally, ID[rd] – the faithfulness constraint associated with the increasingly abstract UR, /ɒɒ/ – remains at zero without any pressure to increase. This is because no alternation exists for /VVN/ vowels, so faithfulness constraints are not driving their surface realization. In cases like Punjabi, then, when an alternation does not exist but a covert UR is still needed, the MaxLex prior fails to restrict abstraction because minimally abstract URs like /ɑɑ/ and increasingly abstract URs like /ɒɒ/ do not rely on distinct constraint weights to accurately model the data.

Constraints	Type	initial w	final w
ID[nas]	faith.	51.37	0.07
IDFIN[nas]	faith.	44.83	100.00
SPRD-L	mark.	92.83	5.42
*NASOBS	mark.	100.00	100.00
*NASG	mark.	99.48	0.02
ID[nas]/_Ṽ	contfaith.	100.00	100.00
*VVN	mark.	100.00	100.00
ID[rd]	faith.	0.00	0.00
*LOWRD	mark.	100.00	100.00

UR	$\mathcal{P}$
/tɑɑʋɑɑn/	0.5
/tɑɑʋɒɒn/	0.5
/tɑɑʋɑ̃ɑ̃n/	0.0

Table 5: Constraint weights and UR probabilities with abstract URs included

5 Learning via Disparity Minimization

In this section, I propose an update to the MaxLex learner that generates a preference for minimally abstract URs over increasingly abstract alternatives, even when the minimally abstract UR does not outperform the increasingly abstract UR in either its accuracy in modeling the data or its deviation from a prior on constraint weights. Specifically, if the disparity component in equation (3) is added to the objective function, assigning probability to URs that introduce disparities increases the loss. Consequently, abstraction will only be preferred if doing so sufficiently increases the likelihood of observing the data.

$$\mathrm{D}(\mathrm{IO}_j) = \sum_{i=1}^{k_j} \left[\mathbf{1}_{\{s^I_{ij} \oplus s^O_{ij} = \varnothing\}} + \sum_{f \in \mathrm{F}} \mathbf{1}_{\{s^I_{fij} \neq s^O_{fij}\}} \right]^2 \tag{3}$$

As shown in the equation, the disparity value for the jth input-output mapping is computed by summing squared segment-level disparity terms across all k_j aligned segments.

Each term within the summation compares the ith input segment (s^I_{ij}) with the corresponding output segment (s^O_{ij}). Two indicator functions contribute to segment-level disparities: the first returns 1 if exactly one of the two segments is null (i.e., an insertion or deletion has occurred); the second iterates over all features f in the feature set F, returning 1 whenever the corresponding input-output segments differ on that feature. When either s^I_{ij} or s^O_{ij} are null, the second term contributes 0 vacuously, since the null segment has no features over which to compare. In effect, incentivizing the minimization of the disparity bias encourages the learner to acquire input-output mappings with as few differences as possible between corresponding segments. Squaring segment-level disparities before aggregating them results in a quadratic increase of the disparity bias as the number of disparities for a given segment increases, thereby enacting harsher penalties for underlying segments that are increasingly divorced from their realization.

The inclusion of a disparity bias in the learner is motivated by both theoretical assumptions and empirical observations about how underlying representations are selected. From a modeling perspective, the updated learner satisfies Occam's Razor: among competing hypotheses that account equally well for the data, the disparity bias favors the simplest one. In the context of UR selection, increasingly abstract URs introduce additional complexity by requiring more transformations between the underlying and surface forms. In the absence of independent motivation, positing such abstract forms results in unnecessary representational complexity.

Indeed, linguists often assume that URs reflect SRs faithfully unless motivated otherwise (Kiparsky, 1982; Baković et al., 2022). This assumption is formalized in Tesar (2014, p.1) through the principle of surface-orientedness, whereby "disparities between input and output are introduced only to the extent necessary" to satisfy independent grammatical restrictions. Similarly, Prince and Smolensky (1993/2004, p.225–226) propose the Lexicon Optimization Principle, which holds that learners should select URs that result in the most harmonic output, minimizing violations unless a more abstract UR yields a demonstrable advantage. Finally, empirical evidence supports the notion that language learners disprefer abstract URs. As shown by Kiparsky (1973), covert URs are often reanalyzed over time as surface-true by successive generations of learners, suggesting a robust bias in favor of minimizing disparities.

What follows demonstrates the computational success of incorporating the disparity bias into the MaxLex learner. The procedure begins in the same way as MaxLex, with an initial stage of phonotactic learning followed by a morphologically-aware learning stage. Here, as in the previous section, the algorithm is provided with two potential covert URs to consider, /ɑɑ/ and /ɒɒ/. Importantly, these are the only two URs that need to be considered under the present analysis to demonstrate that the model prefers minimal abstraction. That is, if /ɒɒ/ can be ruled out by the disparity bias, any other covert UR with a superset of the disparities of /ɑɑ/ can also be ruled out. In this case, the UR of /VVN/ vowels must be oral to appropriately model the data, and /ɑɑ/ only differs from the surface form [ɑ̃ɑ̃] in its nasality value. As such, any other potential UR that could effectively model the observed Punjabi forms with a sufficiently high likelihood necessarily possesses a superset of the disparities of /ɑɑ/ and will, therefore, be dispreferred by the disparity bias.

The results of the simulation with the updated learner are provided in Table 6. The weights the learner arrives at are almost identical to the weights learned by the original MaxLex learner. The key difference here is the probability given to the three potential URs considered for [tɑɑʋɑ̃ɑ̃n]. Whereas MaxLex assigned equal probability to both covert URs because they model the grammar equally well and minimize the prior to the same degree, the updated learner assigns essentially all of the probability to the minimally abstract covert UR, /tɑɑʋɑɑn/.

In sum, O'Hara (2017) demonstrated that MaxLex effectively constrains UR abstraction in cases where surface alternations are present and potential covert URs do not dif-

Constraints	Type	initial w	final w
ID[nas]	faith.	51.37	0.00
IDFIN[nas]	faith.	44.83	100.00
SPRD-L	mark.	92.83	4.61
*NASOBS	mark.	100.00	100.00
*NASG	mark.	99.48	0.00
ID[nas]/_V	contfaith.	100.00	100.00
*VVN	mark.	100.00	100.00
ID[rd]	faith.	0.00	0.00
*LOWRD	mark.	100.00	100.00

UR	$\mathcal{P}$
/tɑɑʋɑɑn/	1.00
/tɑɑʋɒɒn/	$9e^{-15}$
/tɑɑʋɑ̃ɑ̃n/	$2e^{-15}$

Table 6: Constraint weights and UR probabilities with abstract URs and the DISPARITY bias.

fer in their disparity count (as in Klamath). Incorporating an explicit disparity bias into MaxLex extends its utility by enabling it to constrain unnecessary abstraction in forms that lack alternations but still require a covert UR for an adequate analysis.

6 Traversing the Search Space

The proposed disparity bias in equation (3) is intimately connected to output-driven maps defined in Tesar (2014, 2016). Tesar's framework shows how disparities between underlying and surface forms can be used to organize the space of potential URs in a way that allows the learner to search efficiently and avoid unnecessary computations.

Output-driven phonology imposes entailment relationships on UR-SR mappings based on their disparity profiles. If a UR maps to a given surface form with n disparities, then any UR that maps to that same surface form with a proper subset of those n disparities must also be grammatical. For instance, if the mapping /tɑ/ → [tu] is grammatical, then /to/ → [tu] must also be grammatical because /to/ → [tu] possesses a proper subset of /tɑ/ → [tu]'s disparities. However, this relationship does not hold between URs that have non-nested disparity sets; for example, /ti/ differs from [tu] in two features (e.g., [front], [round]), but /to/ differs in only one ([high]). Because the disparities in /ti/ → [tu] are not a superset of those in /to/ → [tu], no entailment of grammaticality follows between these mappings.

These entailment relationships allow the learner to organize the space of potential URs for a given surface form into a structured lattice (Figure 1), with the fully faithful UR at the top and increasingly abstract URs further down. Each node represents a potential UR, and edges lead to forms lower down in the lattice that differ by one additional disparity. If a UR at some level of the lattice fails to generate the observed SR, then all URs that include a superset of that UR's disparities (i.e., nodes further down the lattice) can be immediately ruled out. This structure allows the learner to efficiently eliminate broad swaths of the search space.

Importantly, the use of output-driven phonology by Tesar (2014, 2016) to structure the space of potential URs is primarily *negative*: it is designed to rule out more abstract URs based on the failure of a less abstract UR – one higher in the lattice – to map successfully to the surface form. It does not address how a learner might efficiently traverse the remaining space of *successful* URs that can generate the correct SR but differ in the number of disparities they require. Consider again the example lattice in Figure 1. If a learner considers /to/ as a potential UR for [tu] and finds that it is successful in modeling the data, no mechanism exists to prevent it from also needing to consider /tɑ/, /tɒ/, /tõ/, or any other potential UR that contains a proper superset of disparities in its /UR/→[SR] mapping to [tu].

I propose extending output-driven phonology in precisely this direction. A learner equipped with the disparity bias outlined in the previous section and a likelihood threshold at which success in modeling the data is 'good enough' can use the lattice structure not only to eliminate chains of incompatible URs, but also to stop searching the space once this likelihood threshold has been reached and further levels of abstraction only trivially improve the likelihood of observing the data.

More precisely, the search for the optimal UR could be conducted serially rather than initializing UR optimization with the full set of potential URs in contention simultaneously. A learner would begin by considering URs with 0 disparities and then move on to generate and consider URs with successively more

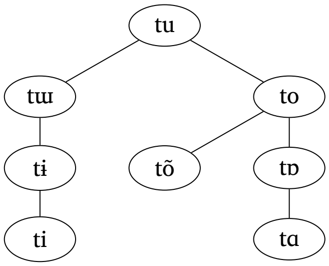


Figure 1: Example lattice for the output form [tu] (c.f. Tesar, 2016)

disparities as needed. As a result, the size of the search space would be irrelevant because the learner does not need to cover the entire space (or even most of it) to decide on the optimal UR.

In sum, the disparity bias does more than minimize abstraction: it also provides a principled way to structure and efficiently search an otherwise infinite space of potential URs. By combining the lattice structure from output-driven phonology with a disparity bias and principled likelihood threshold of acceptability, the framework not only curtails unnecessary abstraction but also offers a computationally efficient method for identifying the optimal UR.

7 Conclusion

This paper introduced a disparity bias as an addition to the MaxLex learner from O'Hara (2017) to improve its preference for minimally abstract underlying representations when multiple URs generate the same surface data with similar likelihood. By penalizing input-output disparities, the model favors URs that more closely resemble their surface realizations, thus curtailing unnecessary abstraction.

In addition to implementing this disparity bias, the paper outlined a blueprint for addressing a second major challenge posed by abstract URs. Specifically, permitting abstraction causes the space of potential URs to grow beyond a size that is computationally feasible to search. Drawing on insights from output-driven phonology, I proposed organizing the UR space into a lattice structured by disparity count and conducting a serial search through this space. By incorporating a likelihood threshold that defines when a UR adequately models the data, the learner can stop the search once candidates with additional disparities fail to meaningfully improve the likelihood of observing data.

While the paper provided a computational implementation of the disparity bias, the proposed method for structuring and traversing the UR space remains conceptual. Future work is required to develop this proposal computationally. This is a non-trivial task. Although concrete URs can be easily identified, generating the set of potential URs for the learner to consider at each increasing disparity level poses a combinatorial challenge. That is, as the number of disparities grows, the number of combined ways in which a segment could be altered to achieve that number of disparities explodes. The matter only worsens when considering multiple segments in a UR. Thus, additional work is needed to determine principled ways to constrain the set of potential URs at each disparity level considered by the learner.

A second open question concerns the likelihood threshold. Although I suggested a threshold as a stopping point, future research must investigate how this value can be grounded empirically. It may be that no single threshold is appropriate across a population of learners, and that the stopping criterion must be calibrated on a speaker-specific basis.

In addition, future work should explore how the disparity bias interacts with the MaxLex prior introduced in O'Hara (2017). This paper has shown that the MaxLex prior alone is insufficient for limiting abstraction in the case of Punjabi pre-N vowels. However, the prior remains crucial in cases like Klamath, where multiple URs generate the same surface form with equivalent disparity counts. Thus, it should be examined whether the disparity component and the MaxLex prior ever conflict, and if so, how such conflicts would be resolved in the learning process.

Finally, the disparity bias was implemented on data from Punjabi, but its application to phonological patterns from other languages that require varying degrees of abstraction is necessary. The cases discussed in Wang and Hayes (2025) would be an interesting set of case studies to begin with in this regard.

References

Adam C. Albright. 2002. *The identification of bases in morphological paradigms*. Ph.D. thesis, UCLA.

Eric Baković, Jeffrey Heinz, and Jonathan Rawski. 2022. *Phonological abstraction in the mental lexicon*. Oxford Academic.

Elena Bashir and Thomas J. Conners. 2019. *A descriptive grammar of Hindko, Panjabi, and Saraiki*. Mouton-CASL Grammar Series. De Gruyter Mouton.

Ivy Hauser and Coral Hughto. 2020. Analyzing opacity with contextual faithfulness constraints. *Glossa: a journal of general linguistics*, 5(1):1--33.

Bruce Hayes and Colin Wilson. 2008. A maxiumum entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry*, 39(3):379--440.

Gaja Jarosz. 2006a. *Rich lexicons and restrictive grammars - Maximum likelihood learning in Optimality Theory*. Ph.D. thesis, Johns Hopkins University.

Gaja Jarosz. 2006b. Richness of the base and probabilistic unsupervised learning in Optimality Theory. In *Proceedings of the Eighth Meeting of the ACL SPecial Interest Group on Computational Phonology*, pages 50--59.

Gaja Jarosz. 2015. Expectation driven learning of phonology. University of Massachusetts manuscript.

Gaja Jarosz. 2019. Computational modeling of phonological learning. *Annual Review of Linguistics*, 5:67--90.

Michael Kenstowicz and Charles Kisseberth. 1977. *Topics in Phonological Theory*. Academic Press.

Michael Kenstowicz and Charles Kisseberth. 1979. *Generative phonology: description and theory*. New York: Academic Press.

Paul Kiparsky. 1973. *Abstractness, opacity, and global rules*, pages 57--86. Tokyo: TEC.

Paul Kiparsky. 1982. *How abstract is phonology?*, chapter 6. Foris Publications.

John J. McCarthy. 2000. Harmonic serialism and parallelism. In *Proceedings of the 30th meeting of the North East Linguistic Society*, pages 501--524.

John J. McCarthy. 2007. *Hidden Generalizations: Phonological Opacity in Optimality Theory*. Sheffield: Equinox.

Charlie O'Hara. 2017. How abstract is more abstract? learning abstract underlying representations. *Phonology*, 34:325--345.

Jonathan Charles Paramore. 2023. Covert URs: evidence from Pakistani Punjabi (talk). In *Formal Approaches to South Asian Languages (FASAL) 14*.

Joe Pater, Robert Staubs, Karen Jesney, and Brian Smith. 2012. Learning probabilities over underlying representations. In *Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology*, pages 62--71.

Alan Prince and Paul Smolensky. 1993/2004. *Optimality Theory: constraint interaction in Generative grammar*. Malden, Mass: Blackwell Publishers.

Bruce Tesar. 2014. *Output-Driven Phonology*. Cambridge: Cambridge University Press.

Bruce Tesar. 2016. Phonological learning with output-driven maps. *Language Acquisition*, 24(2):148--167.

Rachel Walker. 2003. *Reinterpreting transparency in nasal harmony*, pages 37--72. Amsterdam: John Benjamins.

Yang Wang and Bruce Hayes. 2025. Learning phonological underlying representations: the role of abstractness. *Linguistic Inquiry*.

A Appendix

(5) Constraints used in Modeling Punjabi

i. SPRD-L[nas] (cf. Walker, 2003, 47)
For every occurrence of a [+nas] feature in a prosodic word, if that [+nas] feature is dominated by some segment, assign a violation for every segment to the left of that segment in the prosodic word that does not dominate the [+nas] feature.

ii. *NASOBS (Walker, 2003, 51)
Assign a violation for every obstruent that dominates a [+nas] feature.

iii. *NASG (Walker, 2003, 51))
Assign a violation for every glide that dominates a [+nas] feature.

iv. ID[nas]
For every segment, *A*, assign a violation if the output value for the [nas] feature dominated by *A* does not match the input value for the [nas] feature dominated by *A*.

v. IDFIN[nas]
For every segment, *A*, assign a violation if the output value for the [nas] feature dominated by *A* does not match the input value for the [nas] feature dominated by *A* in the final syllable of a prosodic word.

vi. *VVN
Assign a violation for every vowel that dominates a [-nas] feature when directly preceding a nasal consonant.

vii. ID[nas]/_V
Let *A* be a segment that occurs before an oral vowel, _V, in the input. Assign one violation if the output correspondent of *A* does not have the same specifications for [nas] as *A*.

viii. ID[rd]
For every segment, *A*, assign a violation if the output value for the [rd] feature dominated by *A* does not match the input value for the [rd] feature dominated by *A*.

ix. *LOWRD
Assign a violation for every vowel that dominates a [rd] feature and a [low] feature simultaneously.

Dimensions of (dis)preference in designing polar answers in American English: A latent class analysis

Ryan Ka Yau Lai[a] & Yan Lashchev[b]
University of California, Santa Barbara
kayaulai@ucsb.edu[a], ydl@ucsb.edu[b]

Abstract

How we answer questions is often affected by whether our response conforms with the bias, or *tilt*, encoded in the question. For example, if we have a 'yes' answer to a negatively-tilted question like *You aren't eating, right?*, we may delay, hedge and explain our answer. We examine these phenomena at scale through the Switchboard Corpus: We determine which aspects of answer design tend to appear together and how this relates to question tilt through latent class analysis. We find three groups of design features that, respectively, challenge assumptions of the question-answer sequence, expand on the answer, and delay presentation of the answer. We also find that answers contradicting the question's tilt are much closer in design to tilt-conforming answers than responses without polarity, though they do disfavour answers that have *none* of the three classes of features. Results support a gradient and multi-dimensional conception of conversational preference.[1]

1 Introduction

Questions are often designed to be biased, or *tilted*, towards certain types of responses (Bolinger 1957, Heritage & C Raymond 2021). For example, *This is true, isn't it?* is tilted towards 'yes', and *This isn't true, is it?* towards 'no'. An answer congruous with the question's tilt promotes solidarity; the opposite answer may threaten it. This is part of a wider phenomenon called *preference* in Conversation Analysis (Pomerantz & Heritage 2012, Nishizaka & Hayano 2015, Pillet-Shore 2017), specifically the *preference for agreement*, a type of *action preference*: Some actions (e.g. answering positively a positively-tilted question) are *preferred actions*, while others (e.g. answering negatively a positively-biased question) are *dispreferred actions*.

Previous research finds that people minimise the face threat in dispreferred responses by designing them to be less direct (Sacks 1987 [2010], Pomerantz 1985). They may **delay** the answer using silence, audible breaths, laughter, or words like *well*, *uh*; **qualify** it using phrases like *I think*, or **explain** the answer. Such answers have *dispreferred turn formats*; by contrast, short and straight answers have *preferred turn formats*. In other words, previous research found that action preference and design preference tend to go together: preferred actions tend to be implemented with preferred turn formats, and vice versa.

Traditionally, these observations come from qualitative analyses of small datasets. Recent quantitative studies both confirm these observations and complicate the picture. Stivers et al. (2009) find that responses that do not really answer the question are produced slower than answers, and tilt-non-conforming answers are slower than conforming ones. Roberts et al. (2015) find that positive answers are only slightly (~55 ms.) faster than negative ones. Robinson (2020a) argues against the claim that 'neutral' yes-no questions, e.g. *Do you have cats?* asked by someone who does not know the answer, prefer 'yes'; instead, both 'yes' and 'no' answers are preferred responses, while conditional ('it depends')-type answers are dispreferred. Kendrick & Torreira (2015) found that longer delays are much more strongly associated with dispreferred turn formats than with dispreferred actions. Kendrick & Holler (2017) found that dispreferred responses to polar questions were 123-165 ms slower than preferred ones (depending on the operationalisation).

Previous studies have not extensively investigated differences between the various strategies for creating

[1] Thanks to Karen Nylund-Gibson and Delwin Carter for help with modelling, Simon Todd for extensive comments on the paper, and John W DuBois, and members of UC Santa Barbara's CEILing group and the California Annual Meeting on Psycholinguistics for additional discussion.

dispreferred turn formats, which may serve different functions and have different relationships with action preference. This may be in part due to sample size limitations, as disentangling the many strategies requires more than the 200 or so question-answer pairs analyzed in previous work (Robinson 2020a, Kendrick & Torreira 2015). This study examines these differences using corpus-based computational methods, leveraging rich annotations available for the Switchboard Corpus (Godfrey, Holliman and McDaniel 1992). Focusing on polar (i.e. yes-no) questions and their answers in American English, we aim to answer:

1. Are there regularities as to how different answer design strategies appear together?
2. If so, how are the different groups of strategies related to action preference?

The first question is answered by sorting answers into classes according to different features of turn design, then examining which features are associated with which classes, using a latent class model (Nylund-Gibson & Choi 2018). The second is answered by predicting class membership from action preference, using tilt-conformity as an auxiliary variable (Asparouhov & Muthén 2014).

2 Data and methodology

2.1 Corpus and extraction of question-answer pairs

This study uses the Switchboard Corpus (Godfrey, Holliman and McDaniel 1992), consisting of American English telephone conversations between strangers on researcher-designated topics. We mainly made use of the annotations made available in XML format through the NXT-format Switchboard Corpus (Calhoun et al. 2010) and the Switchboard dialogue act corpus (SwDA) (Jurafsky, Shriberg & Biasca, 1997), as converted into CSVs in Potts (2011).

The corpus is divided into approximately utterance-sized units called *slash units*. SwDA assigns a dialogue act annotation to each slash unit, e.g. `qy` for polar questions, `ny` for 'yes' answers, etc. Tags are often modified by adding letters followed by `^`, e.g. `^r` means something is a repetition. Unless otherwise specified, when mentioning a tag in this paper, all the modified versions are included. Appendix A lists and defines all the SwDA tags relevant to this paper.

Polar questions were extracted by searching for the tags `qy` and `^g`. For each extracted question, the next turn from a different speaker than the one who produced the question was extracted as the answer. Question-answer pairs where there was a gap of 5 seconds or longer between the question and the answer were excluded, as they are likely to be erroneous. See Appendix B for the treatment of rare edge cases like multiple questions and turn increments. After question-answer pairs were extracted, we determined whether the answer implements a preferred action and detected different answer design features.

2.2 Features of answer design

Before extracting the features of responsive turns, each turn was divided into three parts. The first slash unit to convey the polarity of the answer (generally tagged `ny`, `na`, `aa`, `nn`, `ng`, `ar`, `no`, `am`, `arp`, `nd`) is called the *core* of the answer in this paper. The parts preceding it are *pre-core*, and the parts following it *post-core*. Answers without detectable cores are not considered. An example is given in Table 1.

A	1	# Like Garth Brooks. # /	Question	`qy^d`
B	2	Garth Brooks, {F oh } /	Pre-core	`^h`
	3	yes, # /	Core	`ny`
	4	{D you know } he's fine. # /	Post-core	`sv^e`

Table 1: Examples of pre-core, core and post-core slash units.

Features of the responsive turn considered in this study are divided into two groups: Those preceding the core or concerning the core itself, and those following the core. The following paragraphs describe how the features were extracted. Though many features were extracted based on the literature, only those appearing >5% of the time were included in the final dataset. Full details of the extraction process and excluded features are in Appendix B.

Pre-core/Core features. The **OFFSET** between two turns was calculated by taking the timestamps of the last word of the question and the first word of the answer. Non-linguistic vocalisms at edges of turns are not considered part of the turn in this calculation. This resembles Offset 2 of Kendrick & Torreira (2015). A negative number indicates overlap between the two turns; a positive number indicates a gap.

Fillers and discourse markers were tagged in the corpus (Meteer & Taylor 1995). Features related to these words are detected either directly using those tags, or using the forms of words (since there are missing tags):

- **FILLERS**: either words other than *oh* tagged `{F }`or having the form *uh* or *um*

- **DMOTHER**: discourse markers other than *oh*, tagged `{D }` or with the forms *well* or *you know*.
- **DMOH**: discourse marker *oh*. It is considered separately as it does not serve to delay the answer, but challenges the question's appropriateness and asserts the answerer's epistemic authority (Heritage 1998, 2005).

Other core-delaying features like breath and laughter were excluded as they did not exceed 5%.

Cores were also tagged for whether they are interjection-type – simple, single-word answers that convey polarity and do not grammatically combine with other words – or non-interjection-type ones (**NONINTERJ**) (called *type-nonconforming answers* in G Raymond (2003)). Cores tagged `nn`, `ny`, are treated as interjection-type answers, plus words like *right*, *yeah*, *sure*, *probably, certainly* when standalone; the rest are non-interjection-type answers. Non-interjection answers are mostly repetitional (Heritage & G Raymond 2012, Enfield 2019), repeating words and grammatical structures in the question (B: *Well, do you do any recycling?* A: *Uh, we do here.*). Some are transformative answers (Stivers & Hayashi 2010) which indirectly imply the answer (A: *You use your, your company's?* B: *My husband's*, which implies a positive answer, but rejects the presupposition that the company is owned by B).

Finally, we looked for words and phrases expressing qualification or epistemic downgrade (**DOWNGRADE**), i.e. lowering the answer's confidence, before or at the core:

- Adverbs like *probably*, *somewhat*, *sometimes*, *personally*, *maybe*, *perhaps*;
- Modal auxiliaries like *could*, *might*, *may*;
- Degree adverbs like *really*, *so*, *very*, *too*, *usually*, with a negator (e.g. *Uh not really*);
- Epistemic/evidential verbs like *think*, *believe*, *guess*, *know*, *say*, *feel*, and common paraphrases, based on Cappelli (2007) and Thompson (2002);
- Slash units tagged `^h` (hedge).

Extraction was aided by part-of-speech tagging and dependency parses from spaCy (Honnibal & Montani 2017) with a three-stage process: adverbs and modal auxiliaries were extracted from the corpus, those related to epistemic downgrade were manually chosen, and then the corpus was reprocessed to detect the chosen forms, reducing the possibility of missing forms that were mistakenly tagged. Note that some downgraders act as interjection-type answers alone (Stivers 2022: 95).

Post-core features. A post-core has the feature **SAMEPOLA** if it contains a polarity-conveying dialogue act with the same polarity as the core. It has the feature **COREEXT** if it contains an extension of the core (with the tag `^e`): these are utterances that repeat or qualify the polarity of the answer, but with more complex expressions than the core (e.g. *Yes, I do*.). A post-core has the feature **EXPAND** if it has a statement (with tag `sv` or `sd`) without the modification `^e` – roughly corresponding to turn expansions (Ford 2001, Lee 2015) in Conversation Analysis. Such expansions can include explanations and elaborations of the core, twists on the core, etc.

Features for fillers, discourse markers, and downgrade were also extracted for the post-core (other than *oh*, which has no known consistent post-core function). An additional feature extracted for post-core but not pre-core is **CONJBUT**, consisting of conjunctions *but* and *(al)though*, because they often present information that contrasts with the polarity conveyed by the core, often in order to qualify it.

Feature	**Definition**	**Location**	**Example**
OFFSET	Time (sec.) between question and answer	PreC/C	B: Do you have **kids**? / A: **[offset = 1.794s] I** have three.
FILLERS	Words like *uh* or *um* that fill pauses	Both	**{F Uh, }** we will be.
DMOH	The discourse marker *oh*	PreC/C	**{F Oh, }** I do.
DMOTHER	Discourse markers other than *oh*	Both	**{D Well, }** {F uh, } I have thought about it.
NONINTERJ	Repetitional and transformative answers	PreC/C	B: Is Texas one of them? A: **Texas is not one of them**.
DOWNGRADE	Language for epistemic downgrade	Both	**Probably** not.
SAMEPOLA	Polarity-bearing dialogue act with the same polarity as the core	PostC	No, / **no.**
COREEXT	Extension of the core	PostC	No, / **I'm not. / [sd^e]**
EXPAND	Statements expanding on the core	PostC	Yeah. /**{F Uh, } I understand. [sv]**
CONJBUT	Contrastive conjunctions like *but*	PostC	No, / I don't, / **{C but }** I think I know what it is.
SISR	Self-initiated self-repair	PostC	Yeah, / **[we, + we've]** seen that, / yeah. /

Table 2: Summary of features included in the final modelling, alongside actual examples from the corpus. PreC/C = Pre-core/core, PostC = post-core, Both = both Pre-core/core and post-core.

Unlike the case of pre-core/core, self-initiated self-repair (SISR) appeared in post-core positions >5% of the time, and was therefore included. A post-core has the feature **SISR** if it has either a slash unit with the tag `%` (abandoned utterance), or brackets `[]` which indicate repair in the transcriptions (Meteer & Taylor 1995). Table 2 summarises and exemplifies all the features included the final modelling.

2.3 Determination of tilt-conformity

The biases that the forms of questions impose on the answer are called *conduciveness* (Bolinger 1957, Quirk et al. 1985) or *tilt* (Heritage & C Raymond 2021). Three question design factors determine tilt: syntactic type, polarity of the question, and presence of negative polarity items.

There are three main **syntactic types** of questions: *Inverted questions* (a.k.a. *interrogative-formatted questions*) are those where the auxiliary verb precedes the subject, e.g. in *Are you eating?*, the auxiliary *are* precedes the subject *you*. *Queclaratives* (a.k.a. *declarative-formatted questions*) have the same syntax as a statement (e.g. *So you're eating.*) but serves as a question, sometimes with rising intonation. *Tag questions* consist of a declarative plus a tag that turns it into a question, usually the word *right* or an inverted auxiliary-subject sequence with polarity reversed from the statement, e.g. *You are eating, aren't you?*, where *aren't you* inverses the polarity of *you are*. The three types are largely determined from SwDA tags: inverted questions have unmodified tags, whereas queclaratives take the modifier `^d` and tag questions `^t`. Some exceptions were manually corrected; details are in Appendix B.3.

The **polarity of the question** is in most cases the polarity of the root of the question in a dependency parse: if a negator depends on it, then it is negative, otherwise it is affirmative. For tag questions, the polarity of the question is defined as the polarity of the declarative portion of the question. When a tag question has an auxiliary-subject sequence as the tag, the root is located in the tag rather than the declarative (e.g. the second *are* in *You are eating, aren't you*), so the polarity of the question is the opposite of the root. Details are in the Appendix.

Negative polarity items (NPIs) are words like *at all*, *any*, *yet* etc., which occur only in negative statements and questions, and are usually said to shift the tilt towards 'no' answers (e.g. Heritage & C Raymond 2021).

From the three question design features above, the tilts of the questions were determined following standard overviews (e.g. Heritage & Clayman 2010: 142-143, Pillet-Shore 2017, Stivers 2022: 11). Queclaratives are tilted towards the same polarity as the statement, e.g. *So you're eating?* is biased towards 'yes', *So you're not eating?* towards 'no'. Tag questions are similarly tilted towards the same polarity as the declarative portion of the question. Positive inverted questions are assumed to be biased towards 'yes' answers, e.g. *Are you eating?* is biased towards 'yes', as are negative inverted questions like *Aren't you eating?*. Table 3 summarises this situation. Questions with NPIs are assumed to be negatively-tilted, unless they are found in negative inverted questions.

Type	Pol	Tilt	Example
Inverted	+	yes	Are you fly fishing?
	-	yes	Isn't that correct?
Quecla-ratives	+	yes	Now this is a LeBaron?
	-	no	You can't read labels?
Tag	+	yes	Those are good aren't they?
	-	no	You don't have mountains in Texas, do you?

Table 3: Types of question syntax without NPIs and their associated tilts. Pol = polarity.

Answers were sorted into tilt-conforming polarity (TC), tilt-non-conforming polarity (TNC), and no polarity (NP) by considering the polarity of the answers. Answers with cores tagged `ny`, `na`, `aa`, `sd^m` were considered positive, and those tagged `nn`, `ng`, `ar` were considered negative; these polarities were compared with the tilt of the question to determine tilt-conformity. Those tagged `arp` and `nd` (answers classified by SwDA as dispreferred) were manually annotated for polarity. Answers tagged `no`, `am` were considered NP; they are neither 'yes' nor 'no', e.g. 'maybe' or 'it depends' answers. Answers without any of these dialogue acts were excluded from the sample; they typically involve transformative answers that do not clearly give a 'yes' or 'no', but do not explicitly refuse to provide a polarity like `no`, `am` either.

2.4 Statistical analysis

The statistical approach taken is mixed mode latent class analysis (MMLCA) (Morgan 2015), which combines latent class and latent profile modelling (Nylund-Gibson & Choi 2018) by allowing both categorical and continuous variables. It identifies distinct categories of answer designs, called *latent classes*, in a data-driven way that does not predefine groups. Each latent class has a distinct distribution of feature values, as well as a prior probability

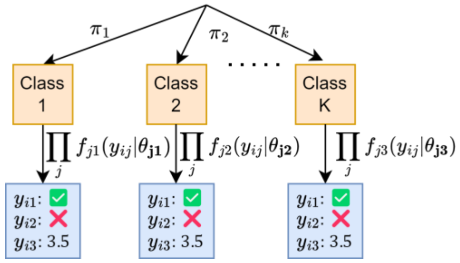

π_1
π_2
π_k
Class 1
Class 2
Class K
$\prod_j f_{j1}(y_{ij}|\theta_{\mathbf{j1}})$
$\prod_j f_{j2}(y_{ij}|\theta_{\mathbf{j2}})$
$\prod_j f_{j3}(y_{ij}|\theta_{\mathbf{j3}})$
y_{i1}: ✓
y_{i2}: ✗
y_{i3}: 3.5

Figure 2: An illustration of the MMLCA for an answer instance with feature profile $\boldsymbol{y_i} =$ [✓, ✗, 3.5], with two dichotomous and one continuous variable.

representing how prevalent it is in the overall corpus. For each answer, the model generates the posterior probability of it belonging to each class, rather than assigning it to a single class. Examining the feature distribution of each class allows us to see and interpret answer designs holistically, abstracting over individual features.

The overall likelihood of the mixed modal latent class analysis model (MMLCA) is:

$$\prod_{i=1}^{N} f(\boldsymbol{y_i}|\boldsymbol{\Phi}) = \prod_{i=1}^{N}\left(\sum_{k=1}^{K}\pi_k\prod_{j=1}^{J} f_{jk}(y_{ij}|\boldsymbol{\theta_{jk}})\right)$$

where $\boldsymbol{y_i}$ is the profile of answer design features like fillers, discourse markers and offset time extracted for answer instance i, $\boldsymbol{\Phi}$ is the model parameters, N is sample size, K is the number of latent classes of answer designs, J is the number of features, π_k is the prior probability of an answer belonging to latent class k, and $\boldsymbol{\theta_{jk}}$ are the class-specific model parameters for the distribution of each feature j in class k. Note that the probability of the features conditional on latent class are multipled together to get their joint probability, i.e. within each latent class, features are assumed independent. For each observation, the most likely latent class is:

$$\underset{1\le k\le K}{\operatorname{argmax}}\left(\pi_k\prod_{j=1}^{J} f_{jk}(y_{ij}|\boldsymbol{\theta_{jk}})\right)$$

After fitting the model, tilt-conformity is used to predict the design of the answer with the ML three-step approach (Vermunt, 2010). The full process is implemented in MPlus (Muthén & Muthén 2019), accessed through `MPlusAutomation` in R (Hallquist & Wiley 2018).

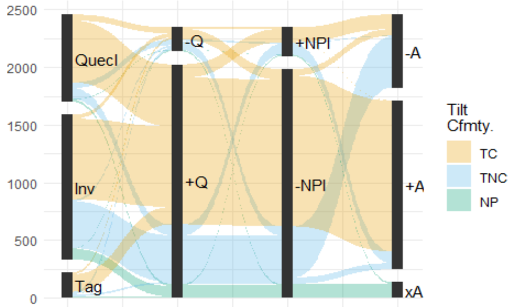


Figure 1: Sankey diagram of extracted data by tilt-related properties. Quecl = queclaratives, Inv = inverted questions, -Q and +Q = negative and affirmative questions, +NPI and -NPI = with and without NPIs, -A and +A = positive and negative answers, xA = no-polarity answers, cfmty = conformity.

3 Results

A total of $N = 2233$ Q-A pairs were extracted from the corpus, slightly more than Stivers' (2022) 1738 and considerably more than most other studies. As shown in Figure 2, there are considerable skews in tilt-related properties: Positive inverted questions without NPIs are by far the most common, followed by positive queclaratives; other categories are much rarer. Other descriptive statistics are in Appendix C; this section will focus on modelling results.

3.1 Latent classes and features

Mixed mode latent class models were run on all the binary turn design features plus OFFSET, which is modelled as Gaussians with class-varying means and variances. Models with 1-7 classes were fitted, with 8000 random starts and 4000 remaining at the final stage. Although different random starts converged to slightly different log-likelihood values, inspection of parameter estimates for top values reveals that they are almost identical.

To find the optimal number of classes, the models with 1-7 classes were compared using a variety of quantitative measures to determine the optimal model, following Nylund-Gibson & Choi (2018). This includes a series of information criteria, plus p-values of the BLRT and VLMR tests, which compare consecutive models: a significant p-value means the more complex model is better than the simpler one (Table 4). After the 5-class model, AWE shows an increase (worsening), and all other information criteria show diminishing returns clearly kicking in at the 6-class model (Figure 3). BLRT is significant for all models; VLMR is insignificant from the 4-class

#C	#Par	LL	BIC	aBIC	CAIC	AWE	BLRT	VLMR
1	15	−15,934	31983	31936	31998	32144	–	–
2	31	−14,327	28893	28794	28924	29225	<0.001	<0.001
3	47	−13,776	27915	27766	27962	28418	<0.001	<0.001
4	63	−13,472	27430	27230	27493	28105	<0.001	0.15
5	79	−13,287	27184	26933	27263	28030	<0.001	0.07
6	95	−13,176	27085	26783	27180	28103	<0.001	0.15
7	111	−13,092	27041	26688	27152	28229	<0.001	0.24

Table 4: #C = Number of classes, #Par = Number of parameters; *LL* = model log-likelihood; BIC = Bayesian information criterion; aBIC = sample size-adjusted BIC; CAIC = consistent Akaike information criterion; AWE = approximate weight of evidence criterion; BLRT = bootstrapped likelihood ratio test *p*-value; VLMR = Vuong-Lo-Mendell-Rubin adjusted likelihood ratio test *p*-value.

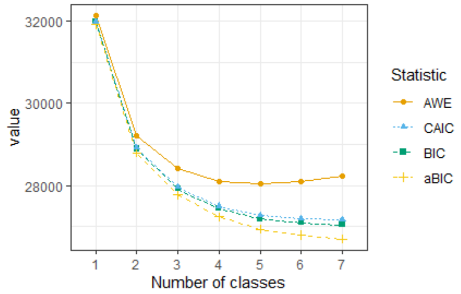


Figure 3: Information criteria for models with varying complexity. AWE worsens and aBIC, BIC and CAIC improve very little after 5 classes.

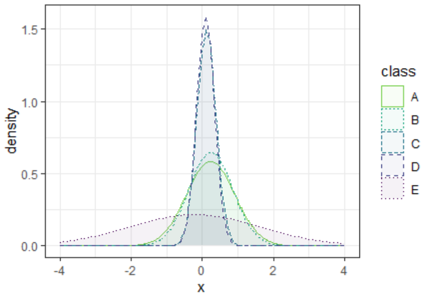


Figure 4: Model-estimated densities of offset values of the five classes. Mean offsets (in sec.) of each class are: A: .098, B: .122, C: .238, D: .249, E: -.205.

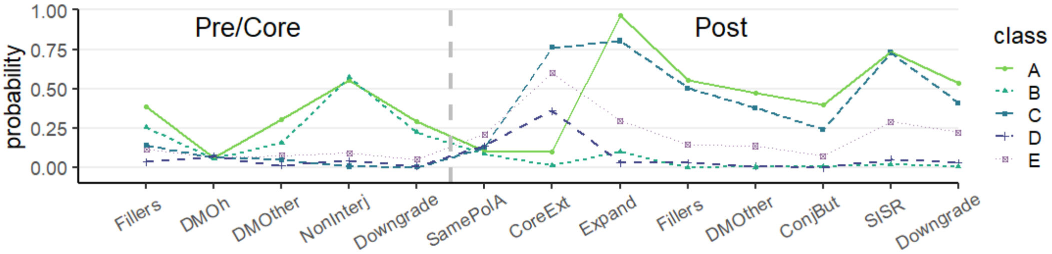


Figure 5: Estimated probabilities of each binary answer design feature by class. The fact that lines cross each other suggests that they play different functions in answer design. If all the features played similar functions and one simply uses more of them if the turn is ‘more dispreferred’, we would expect the lines for different classes to roughly be parallel.

Cl	Description	PreC/C fillers, DMs	Answer type	PreC/C downgrade	Core extension	Post-core expansion & fillers, DMs, etc.
A	Assumption-challenging, strongly delayed & expanded	Most	Both	Many	Very few	Most
B	Assumption-challenging, moderately delayed, unexpanded	Many	Both	Many	None	Little
C	Assumption-conforming, weakly delayed, strongly expanded	Some	Interj.	None	Most	Most
D	Assumption-conforming, undelayed & unexpanded	Few	Interj.	None	Some	Little
E	Unusual offsets	Some	Mixed	Little	Mixed	Mixed

Table 5: The five classes with key properties and brief descriptions of each class. DM = Discourse marker.

model on, though the *p*-value dipped to .07 at the 5-class model. With all metrics considered, we chose the 5-class model.

In the following paragraphs, we will answer our first research question on which answer design features tend to appear together by examining the design feature values associated with each of the five classes.

All five classes' feature profiles (Figure 5 and Figure 6) are amenable to straightforward interpretation. Sample dialogues from each class are in Appendix D. **Class A** contains strongly delayed, hedged, and lengthy answers: these are characterized by the longest offset, are often non-interjection-formatted and downgraded answers, and are most likely to have fillers and discourse markers pre-core as well as expansions and associated features like fillers and discourse markers post-core. **Class B** is like Class A, but with little post-core material and slightly less fillers and discourse markers. Inspection of transcripts also shows that they are mostly transformative, not repetitional answers. **Class C** has much shorter offsets than A-B, many fewer pre-core fillers and discourse markers, and mostly interjection-type answers, but has a similar rate of expansions as Class A. **Class D** has the shortest offsets and least pre-core material, is largely interjection-type, there are some core extensions but almost no expansion. **Class E** has greatest offset variance and largely captures instances with very long gaps or overlaps. In terms of turn design, it only stands out in having the greatest chances of SamePolA, mostly due to turns with long overlaps necessitating repetition; thus, it does not shed much light on the relationship between answer design features, and will not be discussed further in the following paragraphs.

From these observations, we can group features according to the classes they are associated with. Firstly, non-interjection-type cores, pre-core/core epistemic downgrades and lack of core extensions are associated with Class A+B over C+D. These features are ASSUMPTION-CHALLENGING: They convey some stance against what is typically expected of an answer. Epistemic downgrades challenge the assumption that the answerer knows the answer with certainty. Non-interjection-type answers can reject different assumptions, e.g. challenging the relevance of the proposition raised by the questioner, assuming more control over the topics discussed, or increasing one's epistemic authority (Raymond 2003, Enfield et al. 2019, Stivers 2022); this is especially clear in the case of transformative answers, which as mentioned above are most common for Class B. The lack of core extensions is because non-interjection-type answers are already complex and thus hard to extend.

Secondly, post-core expansions and most other post-core features like downgrades, fillers, repair, discourse markers and *but* (which are most likely found in expansions rather than core extensions) are mostly associated with Class A+C over B+D. A+C may be labelled EXPANDED ANSWERS, B+D as NON-EXPANDED ANSWERS.

Finally, pre-core fillers and discourse markers follow the pattern A>B>C>D. These features DELAY the presentation of the answer core. The fact that they differ across all four classes suggests that they serve the double function of anticipating (a) assumption challenges (hence A, B > C, D) *and* (b) a longer, multi-utterance turn (hence A > B, C > D).

Interestingly, offsets pattern primarily with the first group (A, B > C, D), not other delay-related properties, as it is unclear that A>B or C>D. Thus, while our results support Kendrick & Torreira's (2015) suggestion that offset length is an aspect of turn design, silent delays may play a more restricted role than delays with fillers and particles: Longer silence primarily signals assumption-challenging answers, not expanded ones. These differences are small but noticeable: A and D are 151 ms apart.

3.2 Relationship with tilt-conformity

We now proceed to discuss how the various answer design features relate to action preference by examining their relationship with tilt-conformity, under the assumption that tilt-non-conforming answers implement dispreferred actions. Comparing

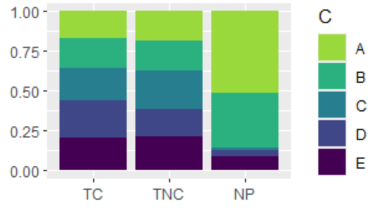


Figure 7: Distribution of probability mass assigned to each class in difference tilt-conformity conditions.

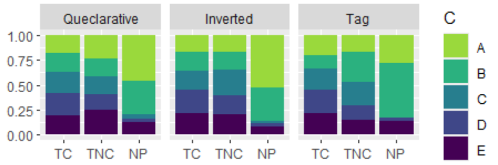


Figure 6: Distribution of probability mass assigned to each class by tilt-conformity and question type.

tilt-non-conforming (TNC) and tilt-conforming (TC) answers, D is much less probable in TNC than TC answer: the odds of getting A, B *and* C over D are higher in TNC answers (A vs D: p = .003; B vs D: p = .005; C vs D: p < 0.001). All other comparisons are insignificant. Comparing non-polarity-bearing (NP) answers to TC ones, the odds of A and B are significantly higher than C, D and E for NP answers (p < 0.001 for all); as is clear in Figure 6, TC-NP differences are much larger than TC-TNC ones, showing that assumption-challenging features are much more associated with NP than turn expansions.

To determine whether this pattern is unique to inverted questions, which dominate the sample, a by-question type barchart is given in Figure 7. The TC-TNC difference is still much smaller than TC-NP or TNC-NP. Because TNC cases are underrepresented, in most cases there is not enough power to quantitatively detect differences between TC and TNC. Visually, however, in tag questions, TNC *may* favour B (assumption-challenging, non-expanded) over not just over D (p = .007) but also C (p = .105) and A (p = .057), suggesting that assumption challenges play a bigger role than expansions in TNC answers to tag questions. However, a larger sample is needed to verify this.

4 Discussion and conclusion

This paper examined turn design in one context: Answers to polar questions in American English, mostly information-seeking questions due to the corpus' nature. We first examined what turn design features tend to go together. Most of the features examined fall into three categories depending on how they co-occur: assumption challenges, answer expansions, and delaying strategies. The three typical sets of strategies traditionally said to characterise dispreferred turn formats (Pillet-Shore 2017) – qualification, accounts (i.e. answer explanations) and delays – fall into these three categories. This suggests that the three types of strategies have distinct distributions and thus functions.

Two unexpected observations emerge from this typology. Firstly, while the choice between interjection- vs. non-interjection-type answers is usually associated with a separate dimension (G Raymond 2003) from the dispreferred turn design strategies of qualification, account and delay, we find that it patterns with qualification in the assumption-challenging category. Indeed, only 5% of interjection-type answers are downgraded, while 21% of non-interjection-type answers are. Secondly, offset patterns with assumption-challenging features rather than other (nonsilent) delay-related features, suggesting that silent delays project only assumption-challenging, not expanded answers.

The fact that nonsilent delays correlate with both assumption challenges and answer expansions may be explained by multiple mechanisms. Firstly, they may anticipate the other turn design features, e.g. Heritage (2015) argues that *well* alerts the listener to upcoming nonstraightforward, transformative *and* expanded answers. They may also directly signal similar meanings as some other answer design strategies, e.g. difficulty in memory retrieval or lower level of knowledge (Smith & Clark 1993, Brennan & Williams 1995), which presumably correlate with epistemic downgrades.

To examine how action preference is related to answer design, we also examined the relationship between tilt-conformity and answer design. As expected, tilt-nonconformity disfavours answers with no delays, expansions, *or* assumption-challenging features over answers with at least some of these. TNC status may favour assumption-challenging features even more in tag questions, probably because they have stronger tilts, and thus going against the tilt poses a greater face threat. Yet, regardless of question type, the tilt-conformity effect is far smaller than the difference between non-polarity-conveying and polarity-conveying answers (regardless of tilt-conformity): Answers without polarity are overwhelmingly designed with non-interjection-type answers and/or epistemic downgrades, likely because they inherently challenge the assumption that the answerer is willing and able to give a straightforward yes/no. This extends Robinson's (2020a) hypothesis that 'yes' and 'no' answers are both preferred answers to positive inverted questions, and only conditional answers are dispreferred, by expanding it to all polar question formats with non-polarity-bearing answers. One difference between Robinson's and our study is that he found no significant difference in pre-beginning behaviour (including fillers and discourse markers in our study) between tilt-conforming and tilt-nonconforming answers, while we do find that tilt-nonconforming answers disfavour class D, which has the least pre-beginning behaviour. This is likely a result of our larger sample size, and supports Robinson's idea that although the *social action* of asking a positive inverted question doesn't by itself impose a preference, the syntactic form still encodes a tilt (Robinson 2020b).

Our results favour a gradient, multidimensional view of preference (Robinson 2020a). Limited by the categories employed by pre-existing SwDA annotations, our study cannot fully examine this richness, e.g. we could not distinguish between expansion types or determine which questions are truly information-seeking. Future studies will hopefully shed further light on these dimensions, a key piece of research as dialogue systems strive to mimic human conversational behaviour (Alloatti et al. 2021, Dingemanse & Liesenfeld 2022, Lah & Lee 2023).

References

Alloatti, Francesca, Luigi Di Caro, Alessio Bosca, & others. 2021. Conversation analysis, repair sequences and human computer interaction–a theoretical framework and an empirical proposal of action. In *Proceedings of the Fourth Workshop on Reasoning and Learning for Human-Machine Dialogues (DEEP-DIAL 2021) at the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*. Association for the Advancement of Artificial Intelligence.

Asparouhov, Tihomir & Bengt Muthén. 2014. Auxiliary variables in mixture modeling: Three-step approaches using Mplus. *Structural Equation Modeling: A Multidisciplinary Journal* 21(3). 329–341.

Bolinger, Dwight. 1957. *Interrogative structures of American English: the direct question* (Publication of the American Dialect Society; 28; No. 28). Alabama: American Dialect Society.

Brennan, Susan E & Maurice Williams. 1995. The feeling of another′ s knowing: Prosody and filled pauses as cues to listeners about the metacognitive states of speakers. *Journal of Memory and Language* 34(3). 383–398.

Calhoun, Sasha, Jean Carletta, Jason M Brenier, Neil Mayo, Dan Jurafsky, Mark Steedman & David Beaver. 2010. The NXT-format Switchboard Corpus: a rich resource for investigating the syntax, semantics, pragmatics and prosody of dialogue. *Language Resources and Evaluation* 44. 387–419.

Cappelli, Gloria. 2007. *"I reckon I know how Leonardo da Vinci must have felt ...": epistemicity, evidentiality and English verbs of cognitive attitude*. Pari (Grosseto): Pari Publishing.

Dingemanse, Mark & Andreas Liesenfeld. 2022. From text to talk: Harnessing conversational corpora for humane and diversity-aware language technology. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5614–5633. Dublin, Ireland: Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.385.

Enfield, N. J., Tanya Stivers, Penelope Brown, Christina Englert, Katariina Harjunpää, Makoto Hayashi, Trine Heinemann, et al. 2019. Polar answers. *Journal of Linguistics* 55(2). 277–304. https://doi.org/10.1017/S0022226718000336.

Fischer, Kerstin. 2015. Conversation, Construction Grammar, and cognition. *Language and Cognition* 7(4). 563–588. https://doi.org/10.1017/langcog.2015.23.

Ford, Cecilia E. 2001. At the intersection of turn and sequence. In Margaret Selting & Elzabeth Couper-Kuhlen (eds.), *Studies in interactional linguistics* (*Studies in Discourse and Grammar* 10). Amsterdam: John Benjamins Publishing. 51–80.

Jurafsky, Daniel, Elizabeth Shriberg & Debra Biasca. 1997. Switchboard SWBD-DAMSL Shallow-Discourse-Function Annotation Coders Manual.

Godfrey, John J, Edward C Holliman & Jane McDaniel. 1992. SWITCHBOARD: Telephone speech corpus for research and development. In *IEEE International Vonference on Acoustics, Dpeech, and Signal Processing*, vol. 1, 517–520. IEEE Computer Society.

Hallquist, Michael N. & Joshua F. Wiley. 2018. *MplusAutomation* : An R package for facilitating large-scale latent variable analyses in Mplus. *Structural Equation Modeling: A Multidisciplinary Journal* 25(4). 621–638. https://doi.org/10.1080/10705511.2017.1402334.

Heritage, John. 1998. *Oh*-prefaced responses to inquiry. *Language in Society* 27(3). 291–334. https://doi.org/10.1017/S0047404500019990.

Heritage, John. 2005. Cognition in discourse. In Hedwig Te Molder & Jonathan Potter (eds.), *Conversation and Cognition*, 184–202. 1st edn. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511489990.009.

Heritage, John. 2015. *Well*-prefaced turns in English conversation: A conversation analytic perspective. *Journal of Pragmatics* 88. 88–104.

Heritage, John & Steven Clayman. 2010. *Talk in action: interactions, identities, and institutions* (Language in Society 38). Chichester Malden: Wiley-Blackwell.

Heritage, John & Geoffrey Raymond. 2012. Navigating epistemic landscapes: Acquiescence, agency and resistance in responses to polar questions. In Jan P. De Ruiter (ed.), *Questions*, 179–192. 1st edn. Cambridge University Press. https://doi.org/10.1017/CBO9781139045414.013.

Heritage, John & Chase Wesley Raymond. 2021. Preference and polarity: epistemic stance in question design. *Research on Language and Social Interaction* 54(1). 39–59. https://doi.org/10.1080/08351813.2020.1864155.

Honnibal, Matthew & Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Kendrick, Kobin H. & Francisco Torreira. 2015. The timing and construction of preference: a quantitative study. *Discourse Processes* 52(4). 255–289. https://doi.org/10.1080/0163853X.2014.955997.

Kendrick, Kobin H & Judith Holler. 2017. Gaze direction signals response preference in conversation. *Research on Language and Social Interaction* 50(1). 12–32.

Lah, Ji Young & Yo-An Lee. 2023. Managing turn-taking through beep sounds by Bixby: Applying conversation analysis to human-chatbot interaction constructions. *Linguistic Research* 40. 61–87. https://doi.org/10.17250/KHISLI.40..202309.003.

Lee, Seung-Hee. 2015. Two forms of affirmative responses to polar questions. *Discourse Processes* 52(1). 21–46. https://doi.org/10.1080/0163853X.2014.899001.

Masyn, Katherine E. 2013. Latent Class Analysis and Finite Mixture Modeling. In Todd D. Litte (ed.), *The Oxford Handbook of Qauntitative Methods in Psychology: Vol 2: Statistical Analysis*. Oxford University Press. https://doi.org/10.1093/oxfordhb/9780199934898.013.0025.

Meteer, Marie & Ann Taylor. 1995. *Dysfluency Annotation Stylebook for the Switchboard Corpus*.

Muthén, Bengt & Linda Muthén. 2019. Mplus: A general latent variable modeling program.

Morgan, Grant B. 2015. Mixed mode latent class analysis: An examination of fit index performance for classification. *Structural Equation Modeling: A Multidisciplinary Journal* 22(1). 76–86. https://doi.org/10.1080/10705511.2014.935751.

Nishizaka, Aug & Kaoru Hayano. 2015. Conversational Preference. In Karen Tracy, Todd Sandel & Cornelia Ilie (eds.), *The International Encyclopedia of Language and Social Interaction*, 1–7. 1st edn. Wiley. https://doi.org/10.1002/9781118611463.wbielsi071.

Nylund-Gibson, Karen & Andrew Young Choi. 2018. Ten frequently asked questions about latent class analysis. *Translational Issues in Psychological Science* 4(4). 440.

Pillet-Shore, Danielle. 2017. Preference Organization. In *Oxford Research Encyclopedia of Communication*. Oxford: Oxford University Press. https://doi.org/10.1093/acrefore/9780190228613.013.132.

Pomerantz, Anita. 1985. Agreeing and disagreeing with assessments: some features of preferred/dispreferred turn shapes. In J. Maxwell Atkinson (ed.), *Structures of Social Action*, 57–101. 1st edn. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511665868.008.

Pomerantz, Anita & John Heritage. 2012. Preference. In Jack Sidnell & Tanya Stivers (eds.), *The Handbook of Conversation Analysis*, 210–228. 1st edn. Wiley. https://doi.org/10.1002/9781118325001.ch11.

Potts, Christopher. 2011. The Switchboard Dialog Act Corpus. Computational Pragmatics. http://compprag.christopherpotts.net/swda.html.

Quirk, Randolph, Sidney Greenbaum, Geoffrey Leech & Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. London: Longman.

Raymond, Geoffrey. 2003. Grammar and social organization: yes/no interrogatives and the structure of responding. *American Sociological Review* 68(6). 939–967. https://doi.org/10.1177/000312240306800607.

Robinson, Jeffrey D. 2020a. Revisiting preference organization in context: a qualitative and quantitative examination of responses to information seeking. *Research on Language and Social Interaction* 53(2). 197–222. https://doi.org/10.1080/08351813.2020.1739398.

Robinson, Jeffrey D. 2020b. One Type of Polar, Information-Seeking Question and Its Stance of Probability: Implications for the Preference for Agreement. *Research on Language and Social Interaction* 53(4). 425–442. https://doi.org/10.1080/08351813.2020.1826759.

Roberts, Seán G, Francisco Torreira & Stephen C Levinson. 2015. The effects of processing and sequence organization on the timing of turn taking: a corpus study. *Frontiers in psychology* SA 2015(6). 509.

Sacks, Harvey. 2010 [1987]. On the preferences for agreement and contiguity in sequences in conversation. In Patrick Griffiths, Andrew Merrison & Aileen Bloomer (eds.), *Language in Use: A Reader*, 8–22. Milton Park: Routledge.

Smith, Vicki L & Herbert H Clark. 1993. On the course of answering questions. *Journal of Memory and Language* 32(1). 25–38.

Stivers, Tanya, Nicholas J. Enfield, Penelope Brown, Christina Englert, Makoto Hayashi, Trine Heinemann, Gertie Hoymann, Federico Rossano, Jan Peter De Ruiter & Kyung-Eun Yoon. 2009. Universals and cultural variation in turn-taking in conversation. *Proceedings of the National Academy of Sciences* 106(26). 10587–10592.

Stivers, Tanya & Makoto Hayashi. 2010. Transformative answers: One way to resist a question's constraints. *Language in Society* 39(1). 1–25. https://doi.org/10.1017/S0047404509990637.

Stivers, Tanya. 2022. *The book of answers: alignment, autonomy, and affiliation in social interaction* (Foundations of Human Interaction). New York, NY: Oxford University Press.

Thompson, Sandra A. 2002. “Object complements” and conversation: towards a realistic account. *Studies in Language* 26(1). 125–163. https://doi.org/10.1075/sl.26.1.05tho.

Vermunt, Jeroen K. 2010. Latent class modeling with covariates: Two improved three-step approaches. *Political analysis* 18(4). 450–469.

Appendices

A Switchboard tags

`qy`	polar question
`ny`	‘yes’ answer
`nn`	‘no’ answer
`ny`	affirmative non-‘yes’ answer
`ng`	negative non-‘no’ answer
`no`	other answer
`nd`	dispreferred answer
`aa`	acceptance
`aap`	partial acceptance
`am`	‘maybe’ answer
`ar`	rejection
`arp`	partial rejection
`h`	hold
`^r`	self-repetition
`^m`	other-repetition
`^e`	expansion
`^g`	tag question
`^h`	hedge
`sd`	statement, not opinion
`sv`	statement, opinion
`{F }`	filler
`{D }`	discourse marker
`{C }`	conjunction
`%`	abandoned utterance
`[]`	repair
`<>`	vocalism

B Details of feature extraction

B.1 Details of extracting question-answer pairs and answer features

Before further processing, any slash unit with + as its dialogue act was merged with the preceding act by the same participant. When there are additional slash units after the first question slash unit of a certain turn (for example, reformulations of the question or turn increments), all slash units up to either the the slash unit right before the start of the next turn or the one right after the start of the next turn were considered, whichever one’s midpoint was closer to the start of the next turn.

The last question slash unit of the question turn was considered in determing question type and polarity. This question was parsed with spaCy. If spaCy identified multiple sentences within the slash unit, then we took the one with a question mark if there is only one such slash unit; we took the longest sentence with a question mark if there were multiple such slash units; and we took the longest sentence if there were no question marks.

The following were treated as potential answer cores: `ny` (yes answers), `nn` (no answers), `na` (affirmative non-yes answers), `ng` (negative non-no answers), `no` (’other answers’), `sd^m` (repetition of the other’s question, which generally affirm the answer in this corpus), `aa` and `ar` (acceptance / rejection of question-formatted collaborate completions), plus any `sd` with the word ‘depend’ in it. For each responsive turn, the first slash unit with one of these dialogue acts was treated as core. Some yes/no answers were mistakenly tagged as `b` (backchannels); when they are classified as interjection-type answers (see below) and there are no other slash units in the response, they are treated as ‘yes’ answers. Although `sv` and `sd` often also implemented polar answers, they were not included as it is difficult to automatically determine whether they bear polarity and, if so, whether they are positive or negative. Determination of answer polarity was discussed in the main text.

Well and *you know* were originally extracted separately from other discourse markers, but later merged into the general category.

OFFSET, SISR and NONINTERJ were mostly extracted as stated in the main text; NONINTERJ are those answers classed as `nn` and `ny`. In addition, a small number of answers from other classes were also interjection-type. These were extracted by considering a list of potential interjection-type answers: *yeah*, *no*, *yes*, *uh-huh*, *right*, *huh-uh*, *okay*, *sure*, *exactly*, *absolutely*, *definitely*, *certainly*, *probably*, *yep*, *yip*, *mm-hm*, *of course*, *no question*, *I'll say*, *possibly*, *maybe*, *alright*, *fine*. This list combines the one in Stivers (2022), plus other interjection-type answers fouund in an inspection of all one-word cores attested in the corpus. An answer is considered interjection-type if its core contains one of these interjections alone, or one of these interjections after by *uh, um, oh, well*.

The determination of DOWNGRADE was relatively complex. Lists of adverbs and auxiliaries were created by parsing all the answer (pre-)cores in the corpus, extracting all adverbs and auxiliaries, and determining polarity. Auxiliaries deemed to be downgraders include *could*, *might*, *should*, *may*, *can*, *ought*, *must*. Adverbs deemed to be downgraders on their own were *probably*, *somewhat*, *sometimes*, *personally*, *maybe*, *perhaps*, *possibly*, *fairly*. Adverbs deemed to be downgraders when combined with negation were *really*, *so*, *very*, *too*, *usually*, *exactly, normally*, *particularly, always*; these were only considered downgraders when there is a negator in the same sentence.

Epistemic verbs include the lemmas *think*, *believe*, *guess*, *suppose*, *know*, *feel*, *hear*, *assume*, *bet*, *conjecture*, *consider*, *doubt*, *expect*, *fancy*, *figure*, *reckon*, *gather*, *imagine*, *judge*, *presume*, *sense*, *surmise*, *suspect*, *trust* with *I* as subject, and *say* with subjects other than *I*. Other phrases included were *my guess*, *my feeling*, *I get the feeling*, *looks like.*

B.2 Unused answer design features

The following features were extracted but not used in the end because they appeared less than 5% of the time.

A pre-core/core has the feature HOLD if it contains a slash unit tagged `h` (hold).

Non-linguistic vocalisms are transcribed in the corpus within angular brackets <>. Four were coded into features: Throat-clearing (THROAT) from the tag `<throat_clearing>`, laughter (LAUGH) from the tag `<laughter>`, lip-smacking (LIPSM) from the tag `<lipsmack>`, and breaths (BREATH) from the tag `<breathing>`.

Conjunctions (CONJ) marked `{C }`, with the forms *so*, *but*, *because*, and sentence-initial *And* were treated as conjunctions. Edit terms (EDITTERM) were extracted with `{E }`, with *I mean* originally extracted apart from other edit terms; all edit terms were discarded in the end.

The feature DIFFPOLA was used for dialogue acts conveying a different polarity as the core.

Sure, *exactly* and *really* were considered UPGRADER when not accompanied by negators. *Absolutely*, *definitely* and *certainly* were always considered upgraders.

B.3 Determination of tilt-conformity

Generally, any question without an auxiliary-subject (or copula-subject) sequence or a tag is considered queclarative. This include subclausal questions. The main exception is that when a question omits a copula or auxiliary verb that cannot be omitted in declaratives; in this case, this is considered ellipsis of the beginning of the question (Quirk et al. 1985), e.g. *you got any hobbies that you want to talk about?*. For questions starting with *how about* (e.g. *{C And } how about SILENCE OF THE LAMB? /*), the question type was set to be the same as that of the previous question.

In general, question slash units with `^d` were treated as queclaratives, those with `^g` as tag questions, and other questions were treated as inverted. Sub-clausal questions were treated as declarative. However, there are a number of cases where the Switchboard corpus appeared to use intonation instead of syntax to determine `^d` would be used. To smooth out these inconsistencies, if a question was tagged as inverted but our syntactic parse finds an auxiliary-subject sequence, or the other way around, we manually checked them to determine question type.

Polarity was determined as described in the main text: For all questions but tag questions with auxiliary-subject tags, it was whether the root had a negator dependent; for tags with auxiliary-subject tags, it was the opposite polarity as the tag.

Answer polarity largely was determined as mentioned in the main text. Answers tagged `sd` containing the word *depend* were treated as NP.

C Descriptive statistics

In the main text, we have discussed the model results. In this appendix we present the descriptive statistics to paint a more comprehensive picture of the data.

Relationships among binary turn design features. To examine the relationship between different binary variables, log-odds ratios were computed between each pair of features, and plotted in Figure 8. Positive values mean the features tend to appear together, negative ones mean they tend to appear apart, and zero means no relationship. As is clear from the heatmap, most relationships are non-negative. Most strong positive relationships are concentrated between features of the post-core and, to a lesser extent, between features of the core/pre-core. EXPAND and post-core SISR are especially notable for their strong association with other post-core features, suggesting most of those other features are found in expansions. DMOH, COREEXT and SAMEPOLA are weakly or negatively correlated with other variables, and appear to work independently of other features.

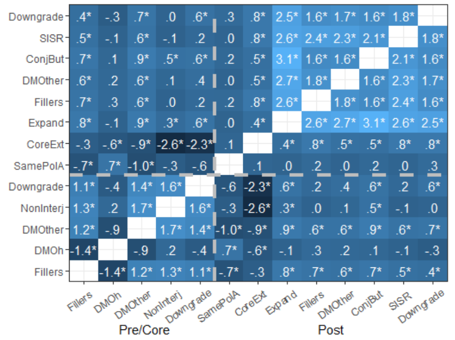


Figure 8: Log-odds ratios between different answer design features. * indicates that the two variables are significantly associated at the .05 level of significance using Fisher's exact tests.

Relationship between OFFSET *and binary turn design features*. Following Kendrick & Torreira (2015), we examine at entire distributions of offsets rather than just means. For each turn design feature, kernel density estimates of the offset were calculated when the feature is present vs when it is absent. The difference between the two densities at various values on (-2, 2) is shown in Figure 9. The clearest pattern is that for all turn design features but DMOH and EXPAND, near-zero (i.e. no gap, no overlap) onsets are much more common when the feature is absent than when it is present. However, the prevalence of gaps over overlaps only seems to be associated with the presence of the pre-core FILLERS and post-core SISR, CONJBUT, and DMOTHER features. For DOWNGRADE and NONINTERJ, longer gaps are associated with the presence of the feature, but so are slight overlaps; only short gaps are associated with absence. For most other features, the pattern is unclear, or even reversed for SAMEPOLA.

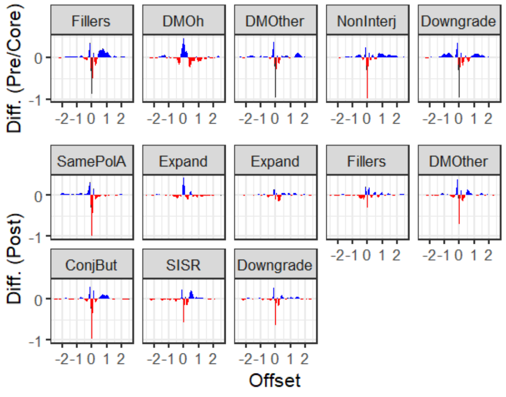


Figure 9: Difference in kernel density estimates of the OFFSET feature when each feature is present vs absent. Red (<0) means that offset value is more common when the feature is absent is larger, and vice versa.

Relationship between tilt-conformity and binary turn design features. Generally, tilt-non-conforming (TNC) turns are more likely to contain the turn design features examined than tilt-conforming (TC) ones, and no-polarity (NP) answers are more likely to contain them than TNC ones, though the degree varies. For pre-core/core NONINTERJ, DMOTHER and DOWNGRADE, the TC-TNC difference is much smaller than the NP-TNC difference; for pre-core FILLERS or post-core STNONEXPAND, the TNC-TC difference and NP-TNC difference are more comparable. DMOH, EXPAND and SAMEPOLA are again exceptions to the general pattern.

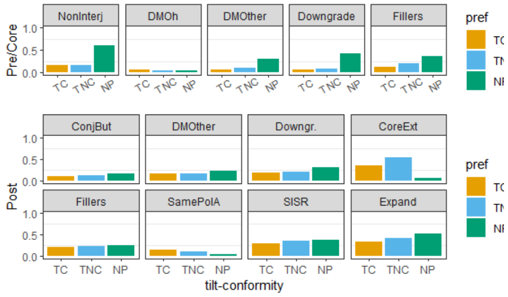


Figure 10: Barcharts of the prevalence of design features in each condition

Relationship between tilt-conformity and offsets. Near-0 offsets are most commonly seen with TC answers, followed by TNC, and finally NP. Gaps between .3-.6 seconds are most likely TNC, followed by NP and TC; beyond around .8 seconds, the order is NP > TNC > TC. From all this, it is clear that NP responses are most closely associated with long gaps, followed by TNC and TC. Nevertheless, the differences are quite minute.

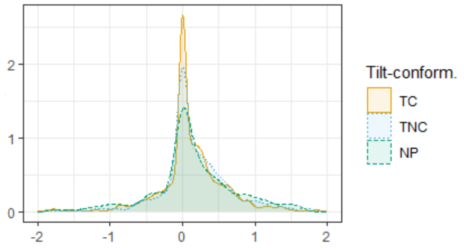


Figure 11: Kernel density of offsets by tilt-conformity.

Zeroing in on inverted questions, we find that positive inverted questions follow the general pattern in Figure 11, but negative questions are radically different: TC (positive) answers actually are *more* likely to have long gaps than TNC (negative) or NP ones (Figure 12). This may be because negative inverted interrogatives still express the speaker's stance that something in the context makes the state of affairs expressed in the question improbable (Heritage & C Raymond 2021).

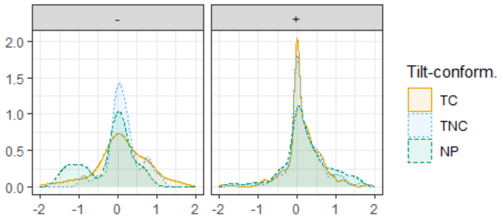


Figure 12: Kernel density estimates of offsets by tilt-conformity for inverted questions without NPIs only.

D Sample answers from the four classes

All answers given in this section have class probability of at least .95.

D.1 Class A

B Is Pennsylvania kind of out of line there? /
A {D Well, } {D actually, } I don't think they're out of line. /
[De-, + Devil's] advocate possibly, /
{C but } <rustling> <inhaling> [it, + you] are trying to avoid paying taxes /
{C and } [whe-, + whether] or not you agree with that law, [i-, + you're] still circumventing it. /
You are legal [in, + in] your circumvention of that law. /

Delays: YES – long.
Fillers and discourse markers: *Well*, *actually.*
Epistemic downgrade: *I don't think.*
Non-interjection answer: Repetitional, not a direct *no.*
Expansion: extensive justification and elaboration after core.

A [You don't, + {F uh, } you're not] [in-, + into] hacking or whatever <laughter>. /
B {F Oh, } [[I, + I think I'm,] + I think I'm] a hacker, /
{C but } I'm [[not, + not kind,] + not [the, + {F uh, } the,]] {D you know, } dial around randomly trying to break into computers type -- -- hackers, /
no, /
that's <laughter> one of those sports I don't go for. /

Delays: YES – long.
Fillers and discourse markers: *Oh, uh, you know.*
Non-interjection answer: Repetition, not a direct *yes*.
Epistemic downgrade: *I think.*
Expansion: extensive justification and elaboration after core.

A {D Well } [don't most of them, + doesn't just] about everything now have both metric and English. /
B They do, /
{C but } things are generally packaged in the English sized packages, {D you know. } /
You buy a quart of milk, /
{C and } sure it [has, + has] the metric equivalent written on there, /
{C but } it still a quart. /

Delays: YES – long.
Fillers and discourse markers: *you know.*
Non-interjection answer: Repetitional, not a direct *yes.*
Expansion: extensive justification and elaboration after core.

D.2 Class B

B Do you have any children? /
A {F Uh, } they're all grown up. /

Delays: YES – moderate.
Fillers and discourse markers: *Uh.*
Non-interjection answer: Transformative, not a direct *no.*
Expansion: NONE, no elaboration or justification after the non-interjection answer.

B Have you read that? /
A {F Uh, } I haven't gotten through <laughter> it yet. /

Delays: YES – moderate.
Fillers and discourse markers: *Uh.*
Non-interjection answer: Transformative, not a direct *no.*
Expansion: NONE, no elaboration or justification after the non-interjection answer.

B Did you all ever watch that? /
A [I, + {D yeah, } I] started, too, and, {F uh } -- -- [kind of, + kind of] worked away from that. /

Delays: YES – moderate.
Fillers and discourse markers: *yeah, uh.*
Epistemic downgrade: *kind of.*

Non-interjection answer: Transformative, not a direct *no.*
Expansion: NONE, no elaboration or justification after the non-interjection answer.

D.3 Class C

A Do you find trouble keeping the records for taxes and all that /
B No, /
it's not hard, /
I just keep it in a notebook and write down what I've made and, {F uh, } {D you know, } what it's going to have to go for that month /
{C and } -- -- {D you know, } it's [not that, + not that] hard. Not at all. /

Delays: YES – minimal.
Fillers and discourse markers: NONE before core.
Interjection-type answer: *No*
Expansion: significant elaboration after the core.

A [Have you, + have you] ever done anything at all? /
B Yeah, /
I have. /
{F Uh, } sit-ups /
{C or, } [al-, + also] last summer I was doing Nautilus /
{C or } last year <cough> I'm, {F uh, } belong to a club right here. /
Got kind of expensive, {F uh, } [to r-, + to [r-, + renew.]] They wanted another fifty dollars. /

Delays: NONE.
Fillers and discourse markers: NONE before core.
Interjection-type answer: *Yeah.*
Expansion: significant elaboration after the core.

A I wonder if she's written anything really recently, if she's got anything [printed, + in print.] /
B Yeah, /
she has, /
{C because } [I, + I] remember seeing a new book by her -- -- that was out, /
{C and } I think [it was a, + it was an] adult book. /

Delays: NONE.
Fillers and discourse markers: NONE.
Interjection-type answer: *Yeah.*
Expansion: significant elaboration after the core.

D.4 Class D

B When you did your papering did you start in the middle of the wall? /
A No /
I didn't. /

Delays: NONE.
Interjection-type answer: *No.*
Expansion: NONE, only extension *I didn't.*

A Have you ever read anything by Susan Howatch? /
B Yes, /
I have. //

Delays: NONE.
Interjection-type answer: *Yes.*
Expansion: NONE, only extension *I have*.

A Like, Queen's Reich, if you ever heard of them. /
B {F Oh, } sure. /
Of course. /

Delays: NONE.
Fillers and discourse markers: *Oh*
Interjection-type answer: *sure.*
Expansion: NONE, only extension *of course*.

D.5 Class E

A {C so. } [Have you, + do you] have a computer for yourself at home? /
B [Offset = 1.21] No /
I didn't. /

Delays: YES – long.
Fillers and discourse markers: NONE.
Interjection-type answer: *No.*
Expansion: NONE, only extension *I didn't.*

B [Do you work with, + do you work around] children when you work? /
A [Offset = -.70] No, /
no, /
not at all. /
I work with <noise> computers. /

Delays: NONE – overlap of speakers.
Fillers and discourse markers: NONE.
Interjection-type answer: *No.*
Expansion: elaboration after the core.

A Do you have any [l-, +] nieces or nephews
<Laughter> ((then)) ? /
B [Offset = -2.09] Yeah. /
Yeah. /
I have a nephew. /
He's a little brat. /

Delays: NONE – overlap of speakers.
Fillers and discourse markers: NONE.
Interjection-type answer: *Yeah.*
Expansion: elaboration after the core

BERT's Conceptual Cartography: Mapping the Landscapes of Meaning

Nina Haket and Ryan Daniels
University of Cambridge
{nch35, rkd43}@cam.ac.uk

Abstract

We present a method for analysing context-sensitive word meanings using BERT embeddings and Gaussian Mixture Models in the fields of lexical pragmatics and Conceptual Engineering. Our methodology generates visual *conceptual landscapes* that reveal how words cluster in different contexts, demonstrated through a case study examining the term PLANET. We provide quantitative metrics for meaning stability and contextual variation, useful for researchers studying lexical pragmatics and meaning change. We also provide an open-source tool which offers an accessible interface for generating visualisations and metrics, requiring minimal technical expertise. Results show that even seemingly straightforward terms exhibit complex meaning landscapes that resist simple definition, highlighting the importance of context-sensitive analyses, combining quantitative metrics and qualitative approaches. This work bridges theoretical pragmatics and computational linguistics, offering empirical grounding for studying how word meanings shift across contexts.

1 Introduction

Language is a complex, dynamic system, constantly evolving and adapting to the contexts in which it is used. Words are not static entities but are deeply embedded in networks of meaning, influenced by both linguistic and extra-linguistic factors. This variability in meaning has long been of interest to linguists, especially in the context of polysemy, the phenomenon of words having multiple related senses (e.g. *paper* as a physical object vs. a scholarly article), and modulations (Recanati, 2010), whereby contextual factors fine-tune a word's interpretation without generating a discrete sense (e.g. an ATM *swallowing* a credit card). We refer to the combination of these polysemous senses and modulation as *contextual meaning variation*, a category encompassing both stable sense multiplicity and more fluid, context-dependent interpretive shifts.

Contextual meaning variations are not merely theoretical concerns – they have significant implications for real-world applications. Conceptual Engineering (CE) is one such domain that directly engages with these issues. CE is concerned with identifying and addressing deficiencies in how words are used, including issues such as vagueness, ambiguity, and biases that distort clear communication (Cappelen and Plunkett, 2020; Cappelen, 2018). Much attention in CE is given to 'improving' words in isolation, but the challenge of modifying word meanings is complicated by the very nature of words: they exist within networks of meanings that shift across different contexts.

In this paper, we propose an interdisciplinary approach that bridges CE, lexical pragmatics, and computational linguistics. We create a tool and method that helps address the practical challenges faced by those navigating the complexities of lexical meaning (e.g. conceptual engineers) by leveraging natural language processing (NLP) techniques to map the intricate relationships within word meanings designed to be broadly useful for researchers in semantics and pragmatics.

Specifically, we use language models such as BERT (Devlin et al., 2019) to generate contextualised embeddings for a selection of words frequently targeted by conceptual engineers, drawn from the spoken component of the British National Corpus 2014 (Love et al., 2017). Using Gaussian Mixture Models (GMMs), we analyze these embeddings to uncover how words cluster in different contextual settings, allowing us to visualise and understand the *conceptual landscapes* of words – how meanings interconnect and shift based on context. These visualisations and metrics map the intricate landscape of meanings associated with a lexical item. Unlike traditional corpus methods such as collocation analyses, our approach condenses embeddings into clear visual representations, highlighting the proximity, distinctness, and relationships between meanings while accounting for contextual and distributional complexities. By mapping the conceptual landscapes of words, we offer lexical pragmaticists and conceptual engineers a way to approach the delicate task of understanding contextual variations with greater precision, while simultaneously advancing the capabilities of NLP research to handle complex, context-dependent word meanings. This includes applications in word sense disambiguation (WSD) and dialogue systems.

2 Related Work

While this tool and methodology have wide-ranging applications, we focus on CE as a case study. CE is inherently practical, aiming to actively modify word meanings rather than merely theorising about them. This dimension makes it even more crucial to have robust methods that allow for precise, context-aware revisions to word meanings, ensuring that any interventions are both effective and sensitive to the complexities of language.

2.1 Conceptual engineering

CE is an emerging area of analytic philosophy concerned with improving the tools we use to think and communicate, namely, our words and concepts, when these are found to be defective in some way (Cappelen, 2018; Koch et al., 2023; Isaac et al., 2022). These 'defects' may be theoretical (e.g. vague, misleading, or imprecise terms) or socio-political (e.g. terms that encode harmful ideologies). A prominent example is Haslanger (2000), who argues that biologically grounded definitions of terms like WOMAN and RACE should be replaced with socially grounded ones to better reflect structural realities and serve emancipatory goals. In this sense, CE is a normative project.

Here, we provide empirical tools that can be used by CE practitioners, and, crucially, also by those who wish to critique or scrutinise their efforts. If CE is to be practised at all, it should be done with a full understanding of how meanings actually function across different contexts of use. This paper seeks to separate diagnosis from prescription, and this is where linguistic analysis has a crucial role to play. We offer a method for mapping the actual complexity of word usage, making it possible to ask more informed questions about what kind of change is feasible, who it affects, and where resistance might arise. For a more nuanced discussion of these facets, see Haket (forthcoming). In this sense, the framework is not a blueprint for linguistic intervention, but a diagnostic system for meaning dynamics.

2.2 Lexical Pragmatics

Lexical pragmatics is concerned with how meaning is shaped by context, particularly the influence of pragmatic factors such as speaker intent, discourse context, and social norms. Meaning can vary significantly across different contexts, with words taking on multiple meanings depending on their use. Polysemy has been a key focus in pragmatics, with scholars like Grice (1989) and relevance theorists (Wilson and Sperber, 2006) exploring how contextual cues guide these inferences on the utterance level, and lexical semanticists/pragmaticists exploring the potential for these contextual meaning variations on a lexical level (e.g. Del Pinal 2015).

CE has often treated meanings as fixed, dictionary-style entries that can be revised in isolation (Cappelen, 2018). However, psycholinguistic research shows that understanding speaker meaning in everyday discourse frequently bypasses full semantic decoding (Gibbs, 1984; Gibbs and Moise, 1997; Bezuidenhout and Cutting, 2002). This suggests that CE should shift its focus from static semantic definitions to the dynamic, context-sensitive meanings that arise in real-world use (Pinder, 2020). However, these present a fundamental challenge that has been undertheorised in the CE literature. Utilising this insight means that conceptual engineers must consider not only stable semantic meanings of words but also the ways in which meaning shifts across contexts, through polysemy or through processes like narrowing, broadening, and metaphorical extension. By incorporating contextual meaning variations into CE, we can more precisely map how word meanings function across discourse and avoid overly simplistic or static revisions

2.3 Computational Lexical Pragmatics

If conceptual engineers indeed need to shift their focus to these lexical pragmatic meanings, they need a way of accessing, analysing, and understanding them. After all, these kinds of meanings may not necessarily appear in dictionaries. The challenge lies in systematically analysing how words are actually used across different contexts, a task that has traditionally been difficult to approach at scale. However, recent advances in computational linguistics, particularly through word embeddings like BERT (Devlin et al., 2019), have revolutionised the study of meaning variation. BERT's contextual embeddings have been shown to capture distributional patterns in language, aligning with the American branch of distributionalism (esp. Harris 1954) that semantically similar words tend to occur in similar contexts (Chiang and Yogatama, 2023; Ferret, 2021). BERT's ability to learn such patterns through its masked language modeling objective has revolutionised our ability to study meaning variation.

More specifically, the clustering and analysis of these kinds of embeddings have led to impressive results in a variety of tasks, particularly WSD (Soler and Apidianaki, 2021). BERT embeddings can capture both contextual variations, with the spatial location of embedded words shifting based on their surrounding context (Coenen et al., 2019), and semantic distinctions between different word meanings and usages (Erk and Chronis, 2022; Chronis and Erk, 2020). This dual capability is supported by multiple empirical findings: embeddings of non-polysemous words show higher similarity than polysemous words (Cevoli et al., 2023; Wilson and Marantz, 2022), and BERT's clustering results correlate strongly with human judgments about meaning similarities (Soler and Apidianaki, 2021). BERT can also capture various other linguistic phenomena including metaphorical uses, syntactic roles, and constructions (Giulianelli et al., 2020).

2.4 Aims of this research

Our work makes a threefold contribution to the field. First, we shift the focus of conceptual engineering from static, dictionary-style definitions to the dynamic, context-dependent variations in meaning that arise in discourse, emphasising the importance of lexical pragmatics for conceptual revision. Second, we apply well-established computational lexical tools, such as embedding and clustering techniques, to conceptual engineering, demonstrating how these methods can identify meanings that need revision based on empirical, context-sensitive data. Third, we provide a practical tool for both conceptual engineers and researchers in lexical pragmatics, enabling the analysis of meaning variation in context and helping to identify inconsistencies or ambiguities. By integrating pragmatic theory with computational techniques, our approach allows for a more systematic analysis of both stable meanings and context-dependent shifts, making the revision process more aligned with pragmatic understanding.

3 Methods

In this section, we present a brief overview of the data used, and the computational methods.

3.1 Data

The Spoken British National Corpus (BNC) consists of 1,251 anonymised, unscripted, face-to-face conversations recorded from 672 volunteers from a range of socioeconomic and demographic backgrounds designed to be a representative sample of the British population (Love et al., 2017). The conversations were collected from 2012 to 2014 in a variety of contexts, including business meetings and radio phone-ins, and therefore are representative of everyday vernacular speech. Work on spoken language is underrepresented in previous empirical work on CE, despite it being the primary mode of communication. As such, we chose to focus our research on this area. The Spoken BNC is released under the Spoken BNC2014 User Licence for non-commercial research and teaching purposes.

3.2 Contextual embeddings

BERT (Devlin et al., 2019) is a widely used transformer-based language model, trained on masked token prediction and next-sentence likelihood. Unlike generative models, BERT is bidirectional, attending to both preceding and following tokens. We use the 336M parameter *bert-large-uncased* model, chosen for its balance of performance, efficiency, and simplicity in analysing semantic meaning in the Spoken BNC. BERT's low-resource, low-complexity nature makes it ideal for researchers with limited computational power, to complete our method in under 24 hours. BERT is released under an Apache 2.0 license.

BERT generates *contextual embeddings*, unique embeddings for each token based on its context, in contrast to *static embeddings* like word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014), which provide a single global representation of a word, ignoring local context. As has been noted, this makes BERT particularly suitable for investigating lexical pragmatic effects: BERT cap-

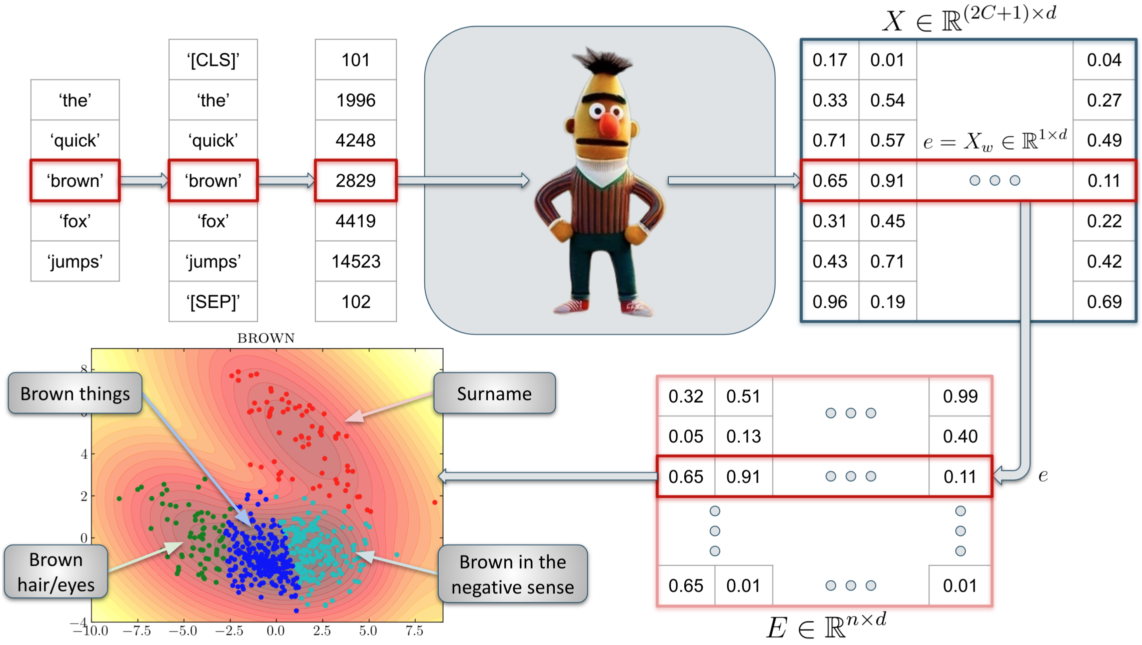


Figure 1: An example of how the target word BROWN is turned into a contextual embedding, e. For a target word the C tokens before and after w are input to BERT. The final embedding e for the target word is then the w^{th} row of the embedding matrix X output from the final hidden layer. A collection of embeddings taken from n sentences are then collated into the matrix E, which is then reduced to 2D and fitted to a GMM.

tures contextual nuances, while static models abstract away this variability.

We generate contextual embeddings for 24 words, *target tokens*, that occur within the Spoken BNC, including words commonly targeted by conceptual engineers such as DUTY, PLANET, TRUTH, and FAMILY (for a full list see Appendix C). These were chosen due to their significance for CE, which usually targets social, moral, political, or philosophical meanings.

We define the context window, C, as half the total number of tokens in the input, excluding the target token, T_w. For a single occurrence of the target token in the text, the total number of tokens fed into BERT is then $2C+1$, where T_w is the middle token: $[T_1, ..., T_C, T_w, T_{C+2}, ..., T_{2C+1}]$. BERT therefore takes as input a $2C + 1$ length utterance. The last layer hidden-state is taken as the output – an embedding matrix $X \in \mathbb{R}^{(2C+1)\times d}$. The word contextual embedding is then the w^{th} row, $e = X_w \in \mathbb{R}^{1\times d}$. For n *separate* occurrences of that target token within the text can be represented by the occurrence matrix $E \in \mathbb{R}^{n\times d}$.

3.3 Conceptual landscapes

A Gaussian Mixture Model (GMM) is a method of modelling multimodal data using a combination of K unimodal distributions. We use a GMM to perform unsupervised soft clustering on the embedding matrix E after dimensionality reduction with principal component analysis (PCA). We optimise K and the number of principal components for each word using the Silhouette score (Rousseeuw, 1987). We then perform a robustness analysis using the Adjusted Rand Index (ARI) (Rand, 1971). The ARI measures the similarity between two sets of cluster assignments. Practically, the ARI ranges between [0,1] with 0 indicating entirely random assignments, and 1 indicating perfect agreement between the two cluster assignments. We fix the number of principal components, and then use 1000 random initialisations for training the GMM. The ARI is calculated for all pairs of cluster assignments for the 1000 random initialisations. We calculate the ARI with (i) 2 principal components, and (ii) the optimal number of principal components. The final labels are calculated by aggregating the results of the 1000 runs into a consensus matrix and using hierarchical clustering on this consensus matrix.

To construct the conceptual landscapes we use the GMM fit to the first two principal components with the optimal number of clusters, and find the log-likelihood scores over a defined space (Figure

1). Limitations and ethical considerations of this methodology can be found in Appendices A and B.

3.4 Metrics

We use four main metrics to describe the landscapes: *maximum explained variance* (MEV), *self-similarity*, *intra-group similarity*, and *inter-group similarity*. The definitions used here closely follow those from Ethayarajh (2019).

MEV If target token T_w appears in sentence i then e_i is the corresponding embedding. The values $\sigma_1, ..., \sigma_m$ are then the first m singular values of the centered occurrence matrix. The MEV is the proportion of variance explained by the first principal component, given by

$$\mathrm{MEV}(w) = \frac{\sigma_1^2}{\sum_i \sigma_i^2} \tag{1}$$

and ranges over $[0, 1]$. MEV indicates the extent to which a contextual embedding could be replaced by a static embedding. If MEV is high it means that the first principal component alone accounts for most of the variation in how a word is used across all its different contexts in the corpus. Even though a given model (such as BERT) produces different embeddings for a word in each context, these embeddings are not scattered randomly. Instead, their differences lie mostly along a single primary axis of meaning variation. You could therefore, in principle, project all the contextual embeddings onto this single line with relatively low loss of information about their overall distribution. A word with a high MEV therefore indicates a uniform consistency of word usage (for example, if the word BARK is always used in the context of "like a dog"). Conversely, if the MEV is low, then no one vector can adequately capture to variation in usage. In terms of CE then, the MEV measures the extent to which changing the semantic meaning is likely to influence speaker meanings.

Self-similarity The self-similarity is the average cosine similarity between embedding vectors, given by

$$\mathrm{Sim}(w) = \frac{1}{n^2 - n} \sum_i \sum_{j \neq i} \cos(e_i, e_j) \tag{2}$$

and ranges over $[0, 1]$. For CE, this metric gives a value of *how much* variation we see within the word. A word with a high self-similarity is constrained in its diversity of usage and meaning, whereas a low self-similarity indicates high diversity in usage.

Anisotropy (the non-uniform distributions of words in embedding space) in LLM contextual embeddings is well documented (Ethayarajh, 2019). It is therefore necessary to control for anisotropy by taking a random sample of embeddings and finding the total average similarity. This baseline is then subtracted from the similarities for each word.

Intra-group similarity Let $e_{k,i}$ be the embedding e_i assigned to label k with n_k members. The global average intra-group similarity for K groups is then

$$\text{Intra} = \frac{\sum_k \sum_i \sum_{j \neq i} \cos(e_{k,i}, e_{k,j})}{\sum_k (n_k^2 - n_k)} \tag{3}$$

For CE, this metric measures similarity within assigned contextual clusters. If the clusters contain contextually similar usages, this score should be high. A high intra-group similarity suggests that the word is used consistently within each cluster, facilitating more precise and effective CE interventions. This allows for targeted modifications to the word's meaning and usage, making it easier to implement changes and achieve the desired conceptual clarity.

Inter-group similarity Let $e_{k,i}$ be the embedding e_i assigned to label k, where n_l are those embeddings *not* assigned to label k. The global average inter-group similarity for K groups is then

$$\text{Inter} = \frac{\sum_k \sum_{l \neq k} \sum_i \sum_j \cos(e_{k,i}, e_{l,j})}{\sum_k \sum_{l \neq k} n_k n_l} \tag{4}$$

For CE, this metric compares members of a single contextual cluster with members from *other* contextual clusters. If the clusters are contextually different from one another, and each individual cluster contains usages which are contextually similar, this score should be low. High inter-group variation suggests more distinct boundaries between contexts, delineating specific usages, which can make CE easier to implement since it can target specific contexts without interference from others.

3.5 Tool

To facilitate practical application of this methodology, we have made a tool publicly available at`https://github.com/acceleratescience/`

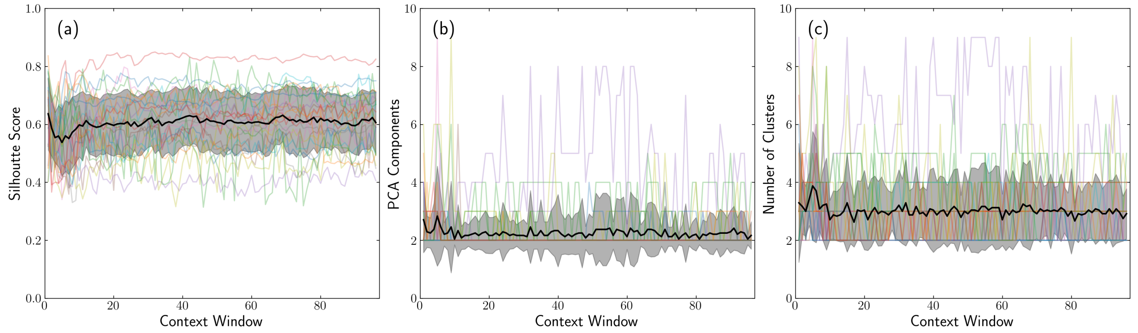


Figure 2: The Silhouette scores (a), optimal number of principal components (b), and optimal number of clusters (c) for each GMM fit to each word. Bold lines indicate averages, and shaded regions indicate the standard deviation.

`conceptual-cartography`. The tool provides an intuitive interface for generating conceptual landscapes and computing the metrics described in this paper. Conceptual engineers can input their target words and corresponding text corpora to visualise meaning clusters, analyze contextual variations, and quantify polysemy through our suite of metrics (MEV, self-similarity, intra-group and inter-group similarity). This enables precise identification of meaning variations and supports evidence-based decision-making in conceptual revision projects. The tool includes comprehensive documentation and example analyses, making it accessible to researchers regardless of their computational background.

4 Results and Discussion

We applied our methodology to a range of words commonly targeted by conceptual engineers, spanning scientific terms (e.g., WEIGHT, ENERGY, PLANET), philosophical concepts (e.g., TRUTH, FREEDOM, KNOWLEDGE), social constructs (e.g., FAMILY, MARRIAGE, EDUCATION), and terms related to technology (e.g., COMPUTER). A complete list of words analysed can be found in Appendix C, and presentation of all the calculated metrics for each word can be found in Table 1 and Table 2.

4.1 Context size

Figure 2 shows the result of optimising the GMM for (a) Silhouette scores, (b) number of principal components, and (c) number of clusters for different context window sizes for the target words. Note that the minimum value of the Silhouette score is achieved at $C = 4$, and therefore when the total number of tokens is ~ 9. The utterance lengths of the Spoken BNC are approximately power-law distributed (see 9) with an average utterance length of ~ 10. This suggests that taking a single utterance as input to BERT may be insufficient to capture the full contextual meaning of the target word. This lends credence to modern approaches to meaning that emphasise meaning across entire discourses as opposed to within a single utterance (Jaszczolt, 2015). As the total number of input tokens exceeds the average utterance length, the Silhouette score increases quickly and remains relatively steady, achieving a maximum at $C \sim 40$.

Importantly, the average number of optimal principal components across words and context windows is ~ 2, and the optimal number of principal components is 2 for every word, except for DUTY, and MARRIAGE. For the following sections, we choose a context window of 40, where the Silhouette score is at a maximum. For all subsequent analyses, the number of clusters is fixed to the optimal number of clusters for each word (for Silhouette scores, optimal principal components and optimal number of clusters for each word, see Figure 8).

4.2 Cluster properties

Figure 3 shows the MEV scores and average self-similarities after correcting for anisotropy (a), and the intra-group similarity and inter-group similarity (b) for the target words. These results are in strong agreement with Ethayarajh that static embeddings would be poor substitutes for the contextual embeddings obtained from BERT. In addition, we also found that a control for anisotropy was not necessary when reducing dimensions.

Figure 3c shows that there is an excellent agreement between the ARI scores when using 2 principal components and when using the optimal number of components, suggesting that the 2D representations capture a substantial amount of the clus-

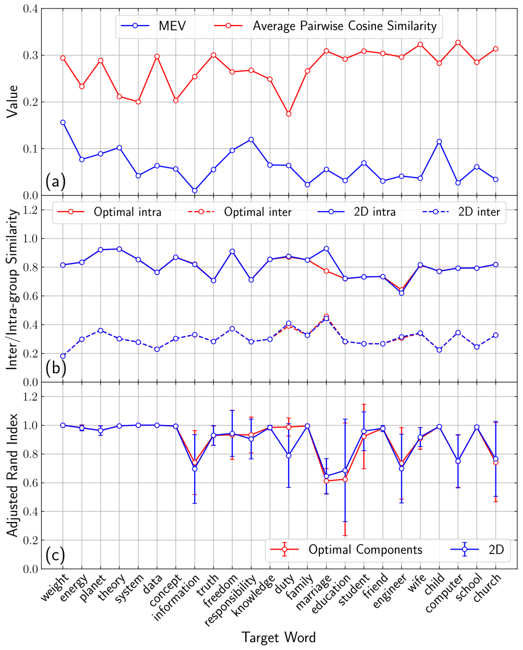


Figure 3: (a) Anisotropy-corrected self-similarity (red) and maximum explained variance (blue). (b) Intra- (solid line) and inter-group (dashed line) similarity for the optimal number of principal components (red), and for 2 principal components (blue). (c) ARI for 1000 GMMs fitted to the optimal number of principal components (red), and for 2 principal components (blue). Error bars are the standard deviations.

tering structure found in the higher-dimensional space. Secondly, the ARI scores show significant variability across words. Words such as WEIGHT, SYSTEM, and FAMILY have high average ARI, and low variance; words such as INFORMATION, EDUCATION, and DUTY, have lower average ARI and higher variance.

Words with high ARIs cluster consistently across different initialisations, indicating a well-defined, stable model, and therefore a well-defined and stable conceptual landscape. The contexts are likely to be more distinct and less ambiguous. Words with lower ARIs may have more ambiguous or varied contexts, causing the clusters to overlap. Therefore, the varying levels of stability reflect the differences between contextual distinctions and ambiguity. The ARI scores for each word are understandably correlated with the Silhouette scores ($r = 0.723$, $p < 0.0001$), given both metrics aim to quantify a measure of cluster quality and stability albeit from different perspectives.

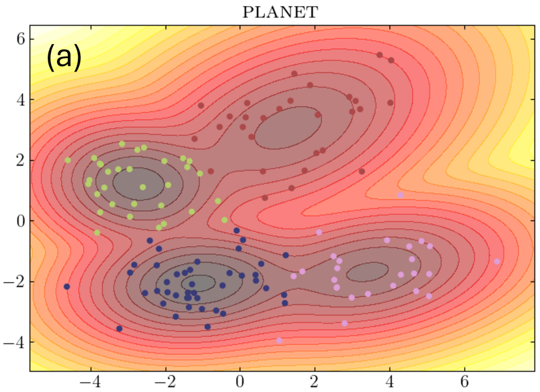


Figure 4: The conceptual landscapes generated using the negative log-likelihood of the GMM predictions in 2D for PLANET with 4 clusters.

4.3 Conceptual landscapes

Since the average number of optimal principal components is approximately 2, it is therefore reasonable to use the 2D conceptual landscape as an indicator of contextual word usage without significant information loss. Figure 4 shows example conceptual landscapes for PLANET (for all target words and landscapes, Figures 6 and 7).

4.3.1 Planet

Due to space constraints and the fact that conceptual engineers typically focus on refining meanings of individual words or closely related sets, this paper analyses a single term (PLANET) to demonstrate how empirical methods can inform CE. The redefinition of PLANET by the IAU in 2006, particularly the exclusion of Pluto, is one of the most frequently mentioned case studies in CE (Landes and Reuter, 2024). Here, it serves here not as a diachronic case study of semantic change, but as a touchstone for the challenges conceptual engineers face when revising the meanings of contextually variable terms. We examine the current semantic landscape in which such revisions take place. Specifically, we ask: when a formal body like the IAU proposes a revision, what kind of semantic structure is it intervening in—and what does that structure imply about the likely uptake, resistance, or diffusion of the revised meaning?

Our analysis reveals both stability and complexity in how PLANET is used. The high ARI of 0.96 indicates consistent, clearly identifiable usage patterns, suggesting distinct meanings that conceptual engineers could potentially target. However, the low MEV of 0.09 demonstrates that no single, static representation can capture the term's full range of uses. The self-similarity score of 0.29, while relatively high, points to considerable contextual variation. Together, these metrics suggest that PLANET exists in a complex semantic space with multiple distinct but related meanings.

This complexity is further illuminated by our identification of four distinct clusters of usage through Gaussian Mixture Model (GMM) analysis and qualitative interpretation:

1. **Astronomical:** Used in scientific contexts to describe celestial bodies in space.
2. **Environmental:** Used in discussions about global ecology or climate change, such as 'saving the planet'.
3. **Metaphorical:** Used to describe a person or object as alien or incomprehensible, as in 'from another planet'.
4. **Hyperbolic:** Used in casual or media contexts to exaggerate the scope of issues or concepts, as in 'worst thing on the planet'.

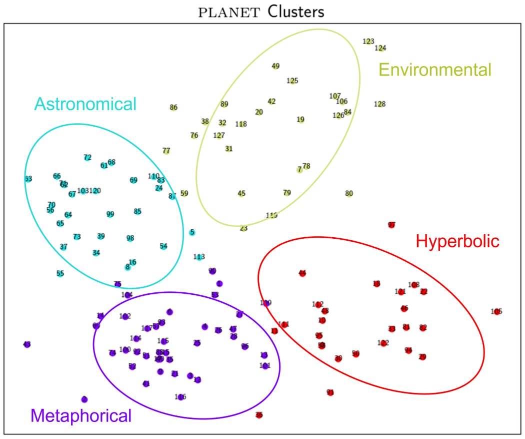


Figure 5: Clusters for PLANET after qualitative analysis.

The PLANETexample illustrates several critical insights for CE.The IAU's redefinition assumes a clear boundary between the astronomical meaning of PLANET and its other uses, such as in environmental or metaphorical contexts. As such, the use of PLANET in environmental contexts ('saving the planet', or even the phrase 'the planet') is of no consequence, as this definition does not depend on whether dwarf planets are PLANETS or not. However, our empirical findings suggest that these meanings are not as easily separated as this theoretical model suggests.

These clusters are not isolated silos: intra-cluster similarity is high (0.92), but inter-cluster similarity remains non-trivial (0.36) indicating gradience and potential overlap between uses. This matters for CE, because it undermines the assumption that a revision to one sense (e.g. the astronomical sense targeted by the IAU) can be neatly isolated from others (e.g. the environmental or metaphorical ones). For instance, even if 'the planet' in 'save the planet' refers to Earth rather than any celestial body, our analysis shows that it remains semantically entangled with the broader category of PLANET.The variability across these different clusters of meaning (especially the overlap between the environmental and metaphorical senses) illustrates the importance of understanding modulation for CE. If conceptual engineers attempt to modify a word's meaning in one context, the resulting revision can inadvertently affect other uses, complicating the task of meaning modification.The observed gradience in meaning—where senses overlap and shift between contexts—illustrates a core challenge for CE. If one sense is revised without accounting for these overlapping uses, unintended consequences may arise in contexts that seem unrelated at first glance, undermining the intended revision.

This complexity is what conceptual engineers must reckon with. Rather than assuming that a term like PLANET can be revised in one domain (e.g. astronomy) without consequence, our data suggests that contextual variations make such revisions porous. In short, if CE is to intervene effectively, it must first understand the semantic terrain it is operating within—and our metrics offer a scalable, replicable way to map that terrain.

4.4 Usage in Conceptual Engineering and Beyond

Conceptual landscapes offer significant theoretical and practical advantages for conceptual engineers. By visualising the variations in meaning of a term like PLANET, conceptual engineers can pinpoint the kinds of meaning they aim to revise and assess how it interacts with other meanings, helping to identify overlaps, dependencies, and links. For instance, revising the astronomical sense of PLANET might clarify scientific discourse, but without careful consideration, it could unintentionally disrupt the metaphorical or environmental uses prevalent in public discussions. These landscapes provide a framework for addressing meaning with precision, sensitivity, and empirical grounding, without requiring extensive training in computational techniques, embeddings, or computer science.

Our methodology offers concrete benefits for CE practice specifically through a structured approach across all stages of the process (see e.g. Koch et al. 2023):

Diagnostic Phase: Identify major meaning clusters, quantify stability (MEV/self-similarity), and map relationships between senses/modulations (inter-cluster similarity).

Planning Phase: Target clusters for revision, predict interference with others, and identify optimal intervention points in the meaning network.

Implementation Phase: Monitor meaning shifts, assess uptake in target contexts, and identify unintended consequences in related clusters.

This framework shifts CE from intuition-based practice to an empirically-grounded methodology, enabling practitioners to visualise and quantify conceptual landscapes. Our tool makes this approach accessible to conceptual engineers without computational expertise, bridging the gap between theoretical CE and practical application. By providing a data-driven understanding of polysemy and variation, it supports both CE and lexical pragmatics. The methodology combines CE's focus on individual words with NLP's large-scale analysis, allowing researchers to explore both the nuances of specific words and broader linguistic landscapes with greater precision.

5 Conclusion

This study introduces a novel methodology for analysing context-sensitive word meanings, bridging the fields of CE, lexical pragmatics, and computational linguistics. First, we have argued for shifting the focus of CE from static definitions to dynamic, context-sensitive meanings. Second, we have provided a methodology for conceptual engineers and lexical pragmaticists to apply computational tools to map the conceptual landscapes of words, revealing polysemy and contextual variations.

As demonstrated through our analysis of PLANET, our approach can effectively identify distinct meaning clusters while quantifying their relationships. The four identified senses (astronomical, environmental, metaphorical, and hyperbolic) and their associated metrics (ARI of 0.96, MEV of 0.09, indicating consistent clustering and strong context-dependence) demonstrate how words can have clearly identifiable yet interrelated meanings that resist simple definition. By leveraging BERT embeddings and Gaussian Mixture Models (GMMs), we generate conceptual landscapes that visualise meaning variation and provide quantitative metrics such as MEV and self-similarity.

Finally, we have created an accessible toolkit that provides a practical and systematic framework for conceptual engineers, linguistic theorists, and others to analyse meaning variation and guide meaning revision efforts, empowering researchers to base their analyses on empirical data rather than abstract intuition.

Acknowledgments

This work was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (www.csd3.cam.ac.uk), provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/T022159/1), and DiRAC funding from the Science and Technology Facilities Council (www.dirac.ac.uk). RKD acknowledges funding from the Accelerate Programme for Scientific Discovery Research Fellowship. NH acknowledges funding from the Cambridge Trust's International Scholarship. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. The views expressed are those of the authors and not necessarily those of the funders. The authors thank Aditya Ravuri, Catherine Breslin, and Kasia Jaszczolt for useful discussions and feedback.

References

Annelie Ädel. 2010. Using corpora to teach academic writing: Challenges for the direct approach. In *Corpus based approaches to ELT*, pages 39–55. Bloomsbury Publishing.

Jack Bandy and Nicholas Vincent. 2021. Addressing" documentation debt" in machine learning: A retrospective datasheet for bookcorpus. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Anne Bezuidenhout and J Cooper Cutting. 2002. Literal meaning, minimal propositions, and pragmatic processing. *J. Pragmat.*, 34(4):433–456.

Herman Cappelen. 2018. *Fixing Language: An Essay on Conceptual Engineering*. Oxford University Press, Oxford.

Herman Cappelen and David Plunkett. 2020. *Introduction: A Guided Tour of Conceptual Engineering and Conceptual Ethics*, pages 1–34. Oxford University Press.

Benedetta Cevoli, Chris Watkins, Yang Gao, and Kathleen Rastle. 2023. Shades of meaning: Uncovering the geometry of ambiguous word representations through contextualised language models. *arXiv preprint arXiv:2304.13597*.

Ting-Rui Chiang and Dani Yogatama. 2023. The distributional hypothesis does not fully explain the benefits of masked language model pretraining. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Gabriella Chronis and Katrin Erk. 2020. When is a bishop not like a rook? when it‘s like a rabbi! multi-prototype BERT embeddings for estimating semantic relationships. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 227–244, Online. Association for Computational Linguistics.

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda B. Viégas, and Martin Wattenberg. 2019. Visualizing and measuring the geometry of BERT. *CoRR*, abs/1906.02715.

G. Del Pinal. 2015. Dual content semantics, privative adjectives, and dynamic compositionality. *Semantics and Pragmatics*, 8(7):1–53.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Katrin Erk and Gabriella Chronis. 2022. Word Embeddings are Word Story Embeddings (and That's Fine). In *Algebraic Structures in Natural Language*. CRC Press.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65. Association for Computational Linguistics.

Olivier Ferret. 2021. Using Distributional Principles for the Semantic Study of Contextual Language Models. In *https://aclanthology.org/events/paclic-2021/*, https://aclanthology.org/events/paclic-2021/, pages 189–200, Shanghai, China.

Raymond W. Gibbs. 1984. Literal meaning and psychological theory. *Cognitive Science*, 8(3):275–304.

Raymond W. Gibbs and Jessica F. Moise. 1997. Pragmatics in understanding what is said. *Cognition*, 62(1):51–74.

M. Giulianelli, M. Del Tredici, and R. Fernández. 2020. Analysing lexical semantic change with contextualised word representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3960–3973. Association for Computational Linguistics.

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

Nina Haket. forthcoming. Navigating meaning spaces: A contextualist approach to conceptual engineering. Manuscript in progress.

Zellig S. Harris. 1954. Distributional Structure. *WORD*, 10(2–3):146–162.

Sally Haslanger. 2000. Gender and Race: (What) Are They? (What) Do We Want Them To Be? *Noûs*, 34(1):31–55.

Manuel Gustavo Isaac, Steffen Koch, and Ryan Nefdt. 2022. Conceptual engineering: A road map to practice. *Philosophy Compass*, 17(10):1–15.

Kasia M. Jaszczolt. 2015. Default Semantics. In Bernd Heine and Heiko Narrog, editors, *The Oxford Handbook of Linguistic Analysis*, pages 743–770. Oxford University Press, Oxford.

Steffen Koch, Guido Löhr, and Mark Pinder. 2023. Recent work in the theory of conceptual engineering. *Analysis*, page anad032.

Ethan Landes and Kevin Reuter. 2024. Conceptual revision in action. Preprint.

Robbie Love, Claire. Dembry, Andrew Hardie, Vaclav Brezina, and Tony McEnery. 2017. The spoken BNC2014: Designing and building a spoken corpus of everyday conversations. *International Journal of Corpus Linguistics*, 22(3):319–344.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha. Association for Computational Linguistics.

Mark Pinder. 2020. Conceptual engineering, speaker-meaning and philosophy. *Inquiry*, pages 1–15.

William M. Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

François Recanati. 2010. *Truth-Conditional Pragmatics*. Oxford University Press, Oxford, New York.

Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

A. Garí Soler and M. Apidianaki. 2021. Let's play mono-poly: Bert can reveal words' polysemy level and partitionability into senses. *Transactions of the Association for Computational Linguistics*, 9:825–844.

Deirdre Wilson and Dan Sperber. 2006. Relevance theory. *The handbook of pragmatics*, pages 606–632.

Kyra Wilson and Alec Marantz. 2022. Contextual embeddings can distinguish homonymy from polysemy in a human-like way. In *Proceedings of the 5th International Conference on Natural Language and Speech Processing (ICNLSP 2022)*, pages 144–155, Trento, Italy. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Limitations

BERT is pretrained on BookCorpus (Zhu et al., 2015) and English Wikipedia (Devlin et al., 2019), which may introduce biases reflective of these contexts into our analysis. By adjusting for anisotropy, we mitigate some of these biases. However, this is not a complete solution. Future work should explore other models and fine-tune on more diverse datasets. In addition, the Spoken BNC includes speech from British individuals over a limited time period, which may not reflect contemporary language use and perspectives, and does not encompass linguistic data from other countries.

While 2D projections are useful for visualising and comparing word contexts, there are instances where higher-dimensional embeddings (e.g., for marriage) provide a clearer representation of semantic differences. This highlights a limitation of our current approach, as projecting down to 2D may obscure important nuances. Future work should explore higher-dimensional embeddings and non-linear dimensionality reduction techniques (e.g., t-SNE, UMAP) to aid visualisation.

Corpus linguistics has been critiqued for its 'inevitable focus on surface forms' (Ädel, 2010), risking an impoverished view of language. We acknowledge this limitation, but argue that CE, being applied and practice-oriented, benefits from observational data on how words are used in context.

B Ethical Considerations

B.1 Use of Language Models

Cultural and language bias. BERT's training data contains cultural biases, including problematic content and skewed religious representation (Bandy and Vincent, 2021). These may affect downstream tasks. Our framework may help identify such biases in training corpora.

The predominance of English in training data limits cultural representation. Fine-tuning on more diverse datasets could mitigate inequities in downstream applications.

Environmental impact. We opted to use BERT for its relative efficiency and smaller environmental footprint, in contrast to larger language models.

Privacy and copyright. While BERT's sources (English Wikipedia, BookCorpus) reduce some privacy concerns, the latter was scraped without author consent, raising ethical issues about data usage.

B.2 Conceptual Engineering

CE attempts to reshape meanings, which can appear overly prescriptive. As meanings are bound to culture and identity, changes not inclusive of diverse perspectives risk alienating the communities they aim to help.

Moreover, CE projects can have social or political ripple effects. We therefore emphasise that this paper offers a descriptive tool: it does not advocate for any particular conceptual change. We provide data about current usage, without prescribing what words should mean.

C Full List of Tested Words

The tested words are:

- *weight*
- *energy*
- *planet*
- *theory*
- *system*
- *data*
- *concept*
- *information*
- *truth*
- *freedom*
- *responsibility*
- *knowledge*
- *duty*
- *family*
- *marriage*
- *education*
- *student*
- *friend*
- *engineer*
- *wife*
- *child*
- *computer*
- *school*

Conceptual landscapes for all words are provided in Figure 6 and Figure 7.

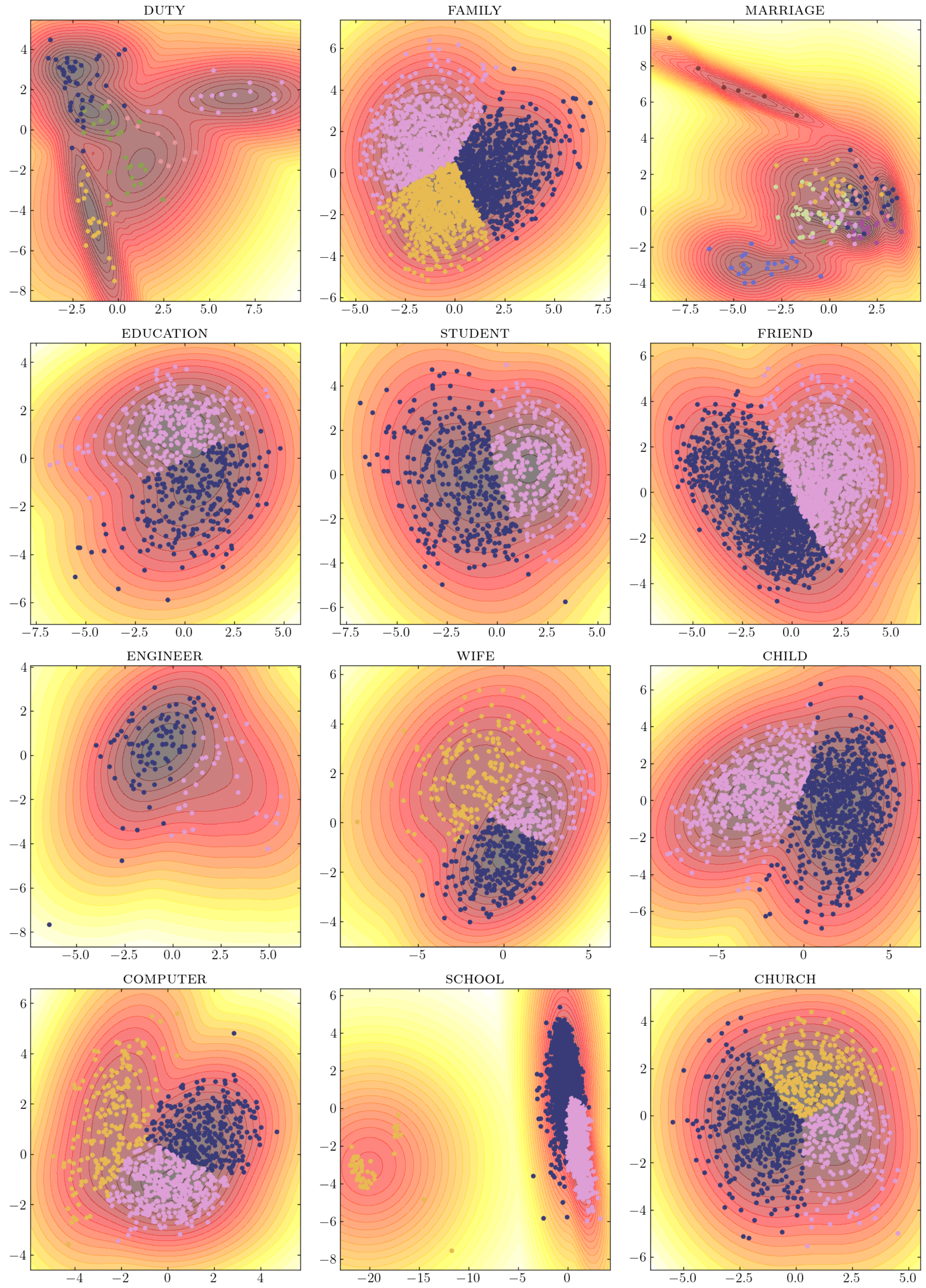


Figure 6: Consensus cluster maps (negative log-likelihood of GMM predictions) for DUTY through CHURCH.

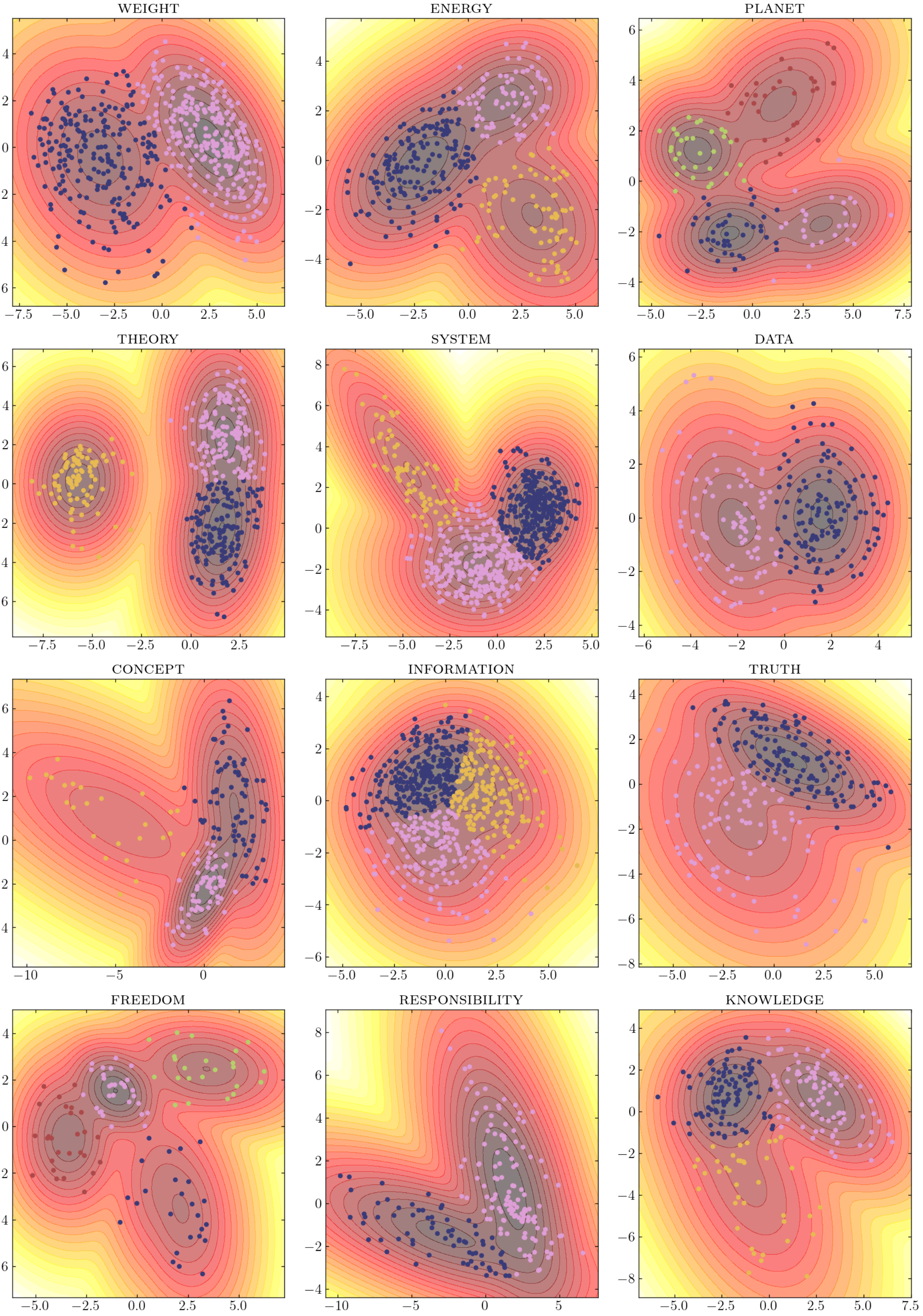


Figure 7: Consensus cluster maps (negative log-likelihood of GMM predictions) for WEIGHT through KNOWLEDGE.

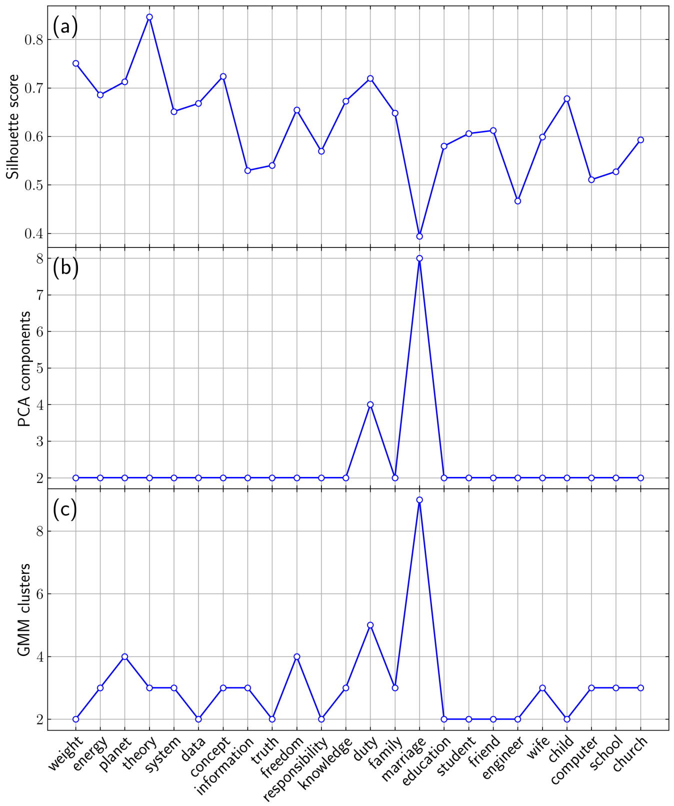


Figure 8: Hyperparameter optimization results: (a) Silhouette scores, (b) number of principal components, (c) number of clusters. Silhouette and ARI scores are closely correlated.

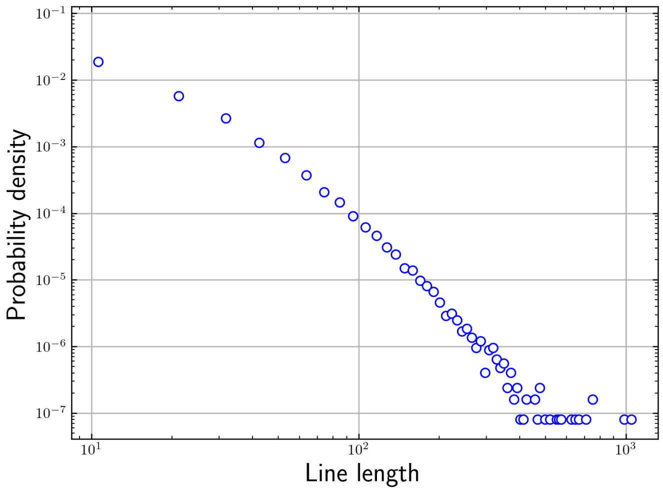


Figure 9: Distribution of utterance lengths in the Spoken BNC. These follow a power-law distribution, with an average of 10 words per utterance.

Word	Optimal Components	Optimal Clusters	Best Score	Self- Similarity	MEV	Optimal ARI	Optimal ARI std	2D ARI	2D ARI std
weight	2	2	0.75065166	0.29663867	0.15589406440022405	1.0	0.0	1.0	0.0
energy	2	3	0.68554884	0.2355035	0.0764136765566644	0.9825799112461127	0.01962176431331328	0.9810035825407845	0.019943568812842878
planet	2	4	0.7128142	0.2913559	0.0882366070740109	0.9623680945153172	0.033088163527237174	0.9626463129788648	0.033075721085404144
theory	2	3	0.84671956	0.21386349	0.10176437079286657	0.9954917857014244	0.004464708197051879	0.9955062177600033	0.004464824823707685
system	2	3	0.6512991	0.20273864	0.04158690442568404	1.0	0.0	1.0	0.0
data	2	2	0.66792816	0.3001163	0.0629035355143655	1.0	0.0	1.0	0.0
concept	2	3	0.723671	0.20564383	0.056091484246831205	0.9927769304661705	0.010173316215442398	0.9935736885883468	0.009859014275290734
information	2	3	0.5296078	0.25662804	0.009946750868021742	0.7406537395085958	0.2229459439592183	0.6960789526143788	0.23915597814367864
truth	2	2	0.54026866	0.30280912	0.054876043198309576	0.9280861918891818	0.06604710754854408	0.9294042503312504	0.06944080370936948
freedom	2	4	0.65434194	0.26655453	0.09563030806831296	0.9334675211205564	0.1704048934448	0.9427797663532299	0.15956748076657506
responsibility	2	2	0.56938255	0.2700225	0.11916091303327889	0.9322590837578956	0.12430886548367147	0.904437876644496	0.13800775307016663
knowledge	2	3	0.6726736	0.25102633	0.0645349155540209	0.9843930515547132	0.015854844628385403	0.9841431866165496	0.015856813383450186
duty	4	5	0.7196939	0.17664373	0.06377767425054486	0.9874551707150581	0.06252875440515283	0.7889657682855794	0.22171666221553094
family	2	3	0.64838034	0.26847154	0.022635997134030736	0.9943836321000173	0.005135851540094211	0.994669251883583	0.004894600248935083
marriage	8	9	0.3940875	0.31149036	0.05500554472960928	0.6112942192156345	0.08721537746728976	0.6447443787878189	0.12265073796622647
education	2	2	0.57998776	0.29428303	0.031460283242946446	0.6239074938781386	0.39106286425586695	0.6854307222784576	0.35662533525437484
student	2	2	0.60584253	0.31139386	0.06900681973577344	0.9219265130861394	0.22414619352034612	0.9590315441198952	0.1349587985075622
friend	2	2	0.6121608	0.30614358	0.03020277056434878	0.9774785414648027	0.021040128884952422	0.9774135569385687	0.02137824727231833
engineer	2	2	0.466553	0.29832488	0.04074953892079064	0.7353913559980679	0.24881847024839246	0.6982006637956051	0.23926154624159524
wife	2	3	0.5990231	0.32535738	0.03624206212147031	0.9089545688639884	0.07546172732778701	0.9178098402172326	0.06638190772251286
child	2	2	0.6781394	0.28536147	0.11498878751245134	0.9901787999403814	0.007877072801540326	0.9904813585767728	0.007709889184639891
computer	2	3	0.5107223	0.32953215	0.026559695009549786	0.7491521491427756	0.18228835360128462	0.7498979023596061	0.18482899770396627
school	2	3	0.52753365	0.28709567	0.060625261536956604	0.9864745788983414	0.011797563212586125	0.9867719126782513	0.011881742870975943
church	2	3	0.5926918	0.31584865	0.03375614676062694	0.7429283349034067	0.2752246758009878	0.7661161198120406	0.25985354840800257

Table 1: Calculated metrics for 24 target words using dimensionality reduction and unsupervised clustering. Metrics include the number of optimal principal components and clusters, best clustering score, self-similarity, maximum explained variance (MEV), ARI scores and standard deviations for both optimal clustering and 2D projections.

Word	Optimal Intra-Sim	Optimal Inter-Sim	2D Intra-Sim	2D Inter-Sim
weight	0.8156033219962284	0.18104519595828528	0.8156033219962284	0.18104515090349865
energy	0.8336862218346286	0.29917733958914533	0.8336863203976584	0.29917730863040515
planet	0.9207608160844708	0.3597553812266942	0.9207608160844708	0.35975542080850764
theory	0.9263468231635071	0.30127710391438284	0.9263466688882307	0.30127714540843425
system	0.8536792740152507	0.27814700771867	0.8536796265171682	0.2781468876547384
data	0.7635351625646621	0.22914614096660874	0.763537767362795	0.2291443617544815
concept	0.8683227585248771	0.30286055940233236	0.8683227585248771	0.30286050245991253
information	0.821972462161749	0.3293009304867715	0.819028850508441	0.3297536590393733
truth	0.7062944748230764	0.28244020454910296	0.7062943393117325	0.28244022355133097
freedom	0.9097507468259896	0.3721182697521081	0.9097507468259896	0.3721182697521081
responsibility	0.7112177734375	0.2816186389568326	0.7112175071022727	0.28161882269883615
knowledge	0.8539526334736376	0.29840904028655746	0.8539525793884304	0.2984091032783977
duty	0.8704321464283045	0.39059547301983527	0.8753667447726858	0.40915019581755635
family	0.8506438458340466	0.32548975138527925	0.850408401614284	0.3256464671847802
marriage	0.7729137680385885	0.4574278943574383	0.9296340574523867	0.443919260225337
education	0.7194930980302446	0.28191425273944803	0.7206263273206777	0.28236683933054896
student	0.7317915722548086	0.2667373108328637	0.7317915722548086	0.26673728025891486
friend	0.7342716988092002	0.2663353340758285	0.7342721319883346	0.2663351627458536
engineer	0.6412440521413054	0.3058740765440698	0.6201696425980734	0.3141900634765625
wife	0.8135365350376823	0.33955498015490126	0.8165262413059602	0.3409786710666057
child	0.7725739291386711	0.2235273103563482	0.7711400170618056	0.22415934626025402
computer	0.7929313357494175	0.3450574308027275	0.7929309680417951	0.34505763451584726
school	0.7931535947179521	0.24504607627722041	0.793153645676212	0.24504624575719003
church	0.818619789088437	0.3265899456336431	0.818619789088437	0.3265899456336431

Table 2: Calculated metrics for 24 target words using dimensionality reduction and unsupervised clustering. Metrics include the Inter-Similarity and Intra-Similarity for both optimal clustering and 2D projections.

On the Dangers of Naïve Replication: The Case of Implicature

Anil Korde
University of Maryland
akorde@umd.edu

Philip Resnik
University of Maryland
resnik@umd.edu

Abstract

Other people's code, data, and definition of a language task often provide the groundwork for new research efforts. The work we present here began as a straightforward investigation of conversational implicature, a central aspect of natural dialogue, starting with updating a prior method to employ more recent LLMs. But differences in results with the work we were replicating led to a deep dive into why those differences were occurring, and this led us to consider more carefully what it means to begin working on a topic with prior work "as a starting point". We describe our process, what we found, and lessons suggested about data quality, task definition, and the current pace of change in NLP.

1 Introduction

Conversational implicature (Grice, 1975) is a ubiquitous phenomenon in conversation, and as such it is highly relevant for conversational AI using large language models. Just as for other language-related capabilities, today's standard paradigm for progress is to use a well defined computational task, together with a benchmark dataset and evaluation metrics, to establish the current state of the art and then adapt or introduce new methods to improve it.

The standard approach is not without its problems, however. Tasks or metrics sometimes turn out to have problems with *measurement validity*, i.e. whether a measurement is actually measuring what we want measured—this has arisen, for example, in natural language inference (Poliak et al., 2018) and topic modeling (Hoyle et al., 2021). Datasets can produce results that don't generalize well. Data contamination may inflate estimates of system performance.

This paper began as an investigation of conversational implicature, aimed at building on prior methods and benchmarking introduced by Ruis et al. (2024). In the end, however, what emerged is a case study contributing to the literature on the pitfalls of uncritically accepting the prompts and data from prior work as a starting point. In the sections that follow, we begin by providing relevant background on the topic of conversational implicature and discuss our attempt to replicate Ruis et al. (2024). We then shift, based on what we found, to a meta-level discussion that leads us to highlight the more general lessons we think this effort turned out to offer about data quality, task definition, and ultimately, we would argue, the pace of change in NLP.

2 Background

The idea of conversational implicature was introduced by Grice (1975). He presents the idea of the Cooperative Principle: that utterances in a conversation are driven by the shared goal of moving the conversation forward. He also states a number of maxims by which the Cooperative Principle is realized. Deliberately violating these maxims, he then argues, is how conversational implicature arises. For instance, in the following exchange, the first speaker's question is not directly answered by the other speaker.

> "Do you want to have dinner tonight?"
> "I have an exam tomorrow."

The plain content of the reply would appear to violate the maxim of Relation ("Be relevant," Grice, 1975). And so the first speaker, upon hearing the reply, is left to infer the meaning that the replier intended to convey by assuming that there is some level at which the maxim is not being violated, even if it appears so at the surface (Levinson, 1983).[1]

[1] A distinction worth noting is that between conversational implicature and conventional implicature. A conversational implicature arises from the context within the conversation in which the utterance is made; in contrast, conventional implicature relies solely on the content of an utterance. A prototypical example of a conventional implicature is the sentence "The

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 199-210.
Eugene, Oregon, July 18-20, 2025

There have been criticisms of Grice's (1975) argument (e.g., Sperber and Wilson (1986) argue that the maxims are so vague as to be unhelpful), but the fundamental point that utterances carry non-conventional meaning is generally accepted. Implicatures and indirect answers of this sort are very common in conversations—occurring in 27% of question/answer scenarios by one account (Rossen-Knill et al., 1997). It follows, then, that large language models trained and productized as chat systems would be more effective if able to use implicature. In addition, users used to human conversation are likely to interact with systems in a way that relies on the system correctly interpreting implicatures in their utterances, even if they do not deliberately set out to do so.

2.1 Prior Work on Implicature

In Louis et al. (2020), a model derived from BERT is trained to predict yes/no answers from a large corpus of indirect question/answer pairs. The authors found that this approach is largely successful, with an accuracy of 80%.

One of the first pieces of research looking at large language models'—rather than models trained specifically for this—ability in this regard is Zheng et al. (2021). The authors introduce a generated dataset of conversations containing implicatures, and then use it to evaluate a number of models' abilities. They note that the use of synthetic datasets if often criticized, and argue that any unnaturalness in their dataset is unrelated to implicatures, since they take care to use "pragmatic phenomena existing in daily conversations" (Zheng et al., 2021).

The BIG-bench benchmarking suite for language models also includes an implicature task (Maru and Bevilacqua, 2022). The authors use a dataset of natural implicatures produced by George and Mamidi (2020), avoiding one of the pitfalls of Zheng et al. (2021). However, Maru and Bevilacqua cut down the dataset by more than half, significantly limiting the size of their analysis.

Hu et al. (2023) look at language models' pragmatic abilities across a number of phenomena, including violations of the Gricean maxims. Performance at answering multiple-choice questions that rely on non-literal understanding is compared across a number of models and with human performance at the same task. They find that the best performing model tested (`text-davinci-002`) performs well above random chance, and often approaches human performance in those tasks. The authors use an expert-curated dataset consisting of 20–40 items per phenomenon. They note that, while this has the significant advantage of being a reliable dataset, its size is a limiting factor.

2.2 Ruis et al. Experiment

In Ruis et al. (2024), the authors look to evaluate the performance of a number of language models at recovering implicatures. They use a dataset of question/response pairs where the responses do not directly answer the question, but carry an implicature. Their experiment takes two forms: looking at the likelihood that the model predicts a 'yes' answer or a 'no' answer in response to an implicature, and a completion-based task where the models are instructed to generate text indicate whether the value of the implicature is yes or no.

For the likelihood task, they give the model a prompt that contains the question, the response, and then establishes a context in which it would be appropriate to output a yes/no answer. Determining whether the model has successfully recovered the correct value of the implicature is done by comparing the likelihoods assigned to the 'yes' and 'no' answers and checking whether the higher likelihood answer matches the implicature value from the dataset. This approach has the advantage of avoiding situations where, if used to generate text, the model would produce output that is neither 'yes' nor 'no,' which would prevent them from easily assessing the model's performance. This has the significant shortcoming, however, that not all models tested provide a way to access the likelihoods of the output. In particular, because some models—such as GPT-3.5-Turbo and GPT-4—are not publicly available (as is the case for a number of the additional models we test in Section 3), the experiments that can be conducted are limited to those that can make use of the online APIs that the developers elect to provide.

For the completion task, Ruis et al. use the same prompts but instead use the model to generate text. If the response ends with the words 'yes' or 'no,' then the responses is considered valid. It's considered correct if the yes/no response matches the

queen is English and therefore brave": the word *therefore* gives rise to the implication that being brave follows from being English (Davis, 2024). This example also highlights the pragmatic phenomenon of *presupposition* (it presupposes that there is currently an English queen), another pragmatic phenomenon that can have important implications (no pun intended!) in LLM-based work (Srikanth et al., 2024).

dataset's value for the implicature of that data point.

They also look at human performance at recovering implicatures in this data set. The same data is given to a group of human annotators who, through an online crowdsourcing platform, are instructed to finish each with 'yes' or 'no' based on what is contextually appropriate. The human annotators achieved an average accuracy of 86%.

Ruis et al. conducted this evaluation comprehensively with 17 different language models, divided into four categories (base models, dialogue fine-tuned, benchmark instruction-tuned, and example instruction-tuned), across 0-shot, 1-shot, and 5-shot scenarios. They find that the models in the Example IT category ("LLMs fine-tuned on tasks with natural instructions for each example," Ruis et al., 2024) consistently perform the best. They also find that, in certain circumstances, the best performing language model (GPT-4) achieves comparable accuracy to the human annotators.

3 Replication

Since Ruis et al. (2024) is one of the more comprehensive pieces of research on language models' performance with implicatures, we began looking into conversational implicature via a very standard approach: replicating the previous findings then seeing whether the results they obtained extend to newer models. We characterize this approach as "naïve" in the sense that it did not involve any particularly careful thought about the actual quality of the previous benchmark in terms of its data or task definition, nor were we particularly concerned with the specifics of the prompts used in the prior work. We simply took the previous benchmark on board uncritically and we assumed that, most likely, advances in language model size and general performance would give us updated baselines to beat.

Our attempt to replicate the results of Ruis et al. (2024) used the same data and a subset of the language models tested there. We also tested several newer models (GPT-4o, Google's Gemini 1.5 Pro, Anthropic's Claude 3, and Meta's Llama versions 3.2 and 3.3) and compared those results. We used the original Ruis et al. (2024) code, adapted for changes in some of the model vendors' APIs.[2] Because, as noted in Section 2.2, the APIs for GPT-3.5-Turbo and GPT-4 (among others) do not provide likelihood information, we only attempted to replicate the completion-based task.

3.1 Modifications

Closely related to prompt engineering, "answer engineering" refers to design choices that facilitate extraction of useful responses from LLM output (Schulhoff et al., 2024). We observed that some original prompts provided LLMs with too much latitude, e.g. "Finish the following text:" when the goal was a yes or no. In order to induce some of the language models (in particular, GPT-3.5-Turbo) to more reliably output yes/no responses as expected by the code, when asked in the 0-shot context for the value of an implicature, we minimally altered some of the prompt templates (see Appendix A): the three original templates which included "Finish the following text:" were modified to read "Finish the following text with yes or no:". This improves the yes/no format consistency of the output; we further modified the Ruis et al. (2024) code to identify the model's answer, not based on the last word of the output, but instead by checking if the response contains, as a whole word, 'yes' or 'no.'

The choice of models was based on those in Ruis et al.'s (2024) Example IT (instruction-tuning) category that were still available. The `text-davinci` models were deprecated by OpenAI in 2024 and are excluded here (OpenAI, 2023a). The Cohere-command-52B (`cohere-command-xlarge`) model is also no longer available; we used Cohere's Command R+ model. The code was extended to allow testing Google and Anthropic models using their APIs, as well as locally-run, open-source models via Ollama.

3.2 Results, Expected...

Table 1 shows the mean and standard deviation in accuracy across the different prompt templates for each of the models tested. For both of the original OpenAI models tested and for all k, accuracy has improved over Ruis et al.'s (2024) results. GPT-4 remains more accurate than GPT-3.5-Turbo though (and is comparable to GPT-4o). Our results also agree with Ruis et al. (2024) that moving from 0-shot to 1-shot to 5-shot does not consistently improve the models' performance.

It is difficult to identify the source of the improvements due to the generally closed nature of the model vendors. But, we expect that the change is likely due to ongoing refinement of the models. For instance, OpenAI notes that they regularly up-

[2] The code can be found on GitHub at `https://github.com/a-korde/llm-implicature-experiment`.

Model	0-Shot	1-shot	5-shot
GPT-3.5-Turbo[3]	77.4% ± 5.9	77.2% ± 4.5	77.6% ± 4.9
GPT-4	86.1% ± 0.7	83.3% ± 0.5	83.9% ± 0.3
GPT-4o	83.1% ± 4.8	84.2% ± 2.9	83.3% ± 2.5
Cohere Command R+	79.8% ± 3.9	80.3% ± 2.6	80.9% ± 1.6
Claude-3.5-Sonnet	**85.6% ± 1.6**	**88.1% ± 1.0**	**89.0% ± 0.6**
Gemini-1.5-Pro	83.5% ± 1.9	84.4% ± 4.2	83.8% ± 4.6
Llama-3.2-3B	60.9% ± 6.5	73.1% ± 13.0	69.9% ± 5.8
Llama-3.3-70B	84.2% ± 1.9	84.9% ± 1.8	84.9% ± 1.2

Table 1: The k-shot accuracy of a subset of the models tested in Ruis et al. (2024), as well as additional models, using our modified prompt templates (see Appendix A). Accuracy is averaged across the different prompt templates.

date models. When these tests were undertaken, the current versions of the OpenAI models used were `gpt-3.5-turbo-0125`, `gpt-4-0613`, and `gpt-4o-2024-08-06`. The Cohere model used was `command-r-plus-08-2024`. The Claude version used was `claude-3-5-sonnet-20241022`. The Gemini version used was `gemini-1.5-pro-002`.

3.3 ...And Unexpected

"The most exciting phrase to hear in science, the one that heralds new discoveries, is not 'Eureka!' but 'That's funny...' "
—Isaac Asimov

We were surprised to see that the one Cohere model tested here showed a dramatic improvement in the 0-shot task over the Cohere-command-52B model tested by Ruis et al., which achieved an accuracy of only 60.2% ± 5.2. One possible explanation for this change was the simple fact that we tested a different model. Changes from the previous Command-52B model's training data or process could have had an impact on its capability in this metric. It would have been fairly natural at this point simply to leave it at that, and move forward with Table 1 as our new baselines—and indeed we considered doing so.

However, Ruis et al.'s (2024) hypothesis about Cohere-command-52B's markedly worse performance on the 0-shot task as compared to the 1- and 5-shot tasks led us to think about an alternative explanation. They hypothesize that the poor 0-shot performance is "not due to a lack of implicature understanding, but due to a failure to calibrate the yes/no likelihoods without examples" (Ruis et al., 2024). That is, they argue the 1- and 5-shot examples serve to clarify the task format and "prime the model towards producing outputs and following the yes/no structure" (Ruis et al., 2024). If that is the case, then our altered prompts (see above) specifically asking for yes/no responses may have contributed to the improved performance.

To test this hypothesis, we re-ran the experiment on Command R+ using the original, unmodified prompt templates from Ruis et al. (2024). In this context, we found that Command R+ performed vastly worse than with our modified prompts. In the 0-shot case, Command R+ had a mean accuracy of just 50.8% ± 48.7 at correctly identifying the value of the implicature. This poor performance, and the very high variability, comes from differing behavior across prompt templates. In three of the original prompt templates—those that were unmodified in our experiment—the model performed in line with our results: it achieved an accuracy of 85.5%, 81.7%, and 72.0% for templates 0, 2, and 3 respectively. With the other three original prompt templates—the ones that we *did* modify—the model performed extraordinarily poorly, with the implicature accuracy varying from 0.8% to 1.5%. The completion accuracy metric (indicating what fraction of the model's generated completions an identifiable answer could be extracted from) shows the same pattern: each of the prompt templates that we did not have to modify all produced usable responses in greater than 98.5% of cases, and those templates that originally used "Finish the following text:" resulted in usable responses in no more than 2.5% of cases.

When given prompts with one example of the task, Command R+'s accuracy jumps to a more expected 73.4%±9.1. The "Finish the following text:" prompts remain somewhat worse performers than the others, however, scoring 62.3%, 65.2%, and 66.0% in implicature accuracy and 83.0%, 87.2%, and 90.8% in completion accuracy. Table 2 gives a

[3] The GPT-3.5-Turbo model is referred to as "ChatGPT" in Ruis et al. (2024).

breakdown of the individual prompt results across $k = 0, 1$ for each of the original and modified prompt templates.

4 Discussion

We viewed the results of our replication attempt as equivocal. On the one hand, we we were able to reproduce the results of Ruis et al. (2024). Frequently, including in our own work, that kind of replication success is sufficient to move on to the more interesting business of trying to build better models and improve the state of the art.

On the other hand, the Asimovian "that's funny" that emerged in our experimentation invited deeper consideration that, we suggest, is more valuable than the replication itself. This is where our discussion pivots from a conversation just about conversational implicature, *per se*, to a reconsideration of the "naïve" approach we took—an approach that is, we would argue, typical of widespread practice in current NLP research—building on a closer look at our replication attempt as a case study.

4.1 Datasets

We begin with data. The experiments here and in Ruis et al. (2024) use a dataset of implicatures in dialogue that have been manually annotated with the value of the implicatures (George and Mamidi, 2020). The data were obtained from two categories of sources: questions from an English language comprehension test (specifically, from free practice versions of the TOEFLS test (English Test Store)) and film scripts from the Internet Movie Script Database (IMSDb). That both of these sources are authored and not naturally occurring could present a difficulty: they may not be representative of how implicatures are used in natural conversation. Movie scripts, in particular, may also be a poor indicator of a model's performance, because the entire script may well have been included in the model's training data.

The dataset's authors also do not go into detail on the labeling process, only noting that "The annotation is done manually by undergraduate students of linguistics, whose primary language of instruction is English" (George and Mamidi, 2020). While the correct answers are provided for the language comprehension test, the same is not true of the entries from movie scripts, and the implicature values provided in the dataset are presumably the judgments of the aforementioned students.

The authors originally intended to crowdsource the dataset of implicatures—going so far as to design and conduct an experiment using an online crowdsourcing platform—but ultimately discarded the data noting that they "did not obtain high-quality dialogue data" (George and Mamidi, 2020). They conclude that the task they designed is somewhat ill-suited to crowdsourcing because it requires more imagination and is less mechanical than is common on crowdsourcing platforms.

This problem is not entirely resolved by using their chosen data sources, though. For instance, the dataset includes an entry with the following context and response utterances, and says that the implicature—the answer to the context question—is 'yes.'

> "Have you found another school for the children?"
> "We're still shopping around."

This does not align with our judgment: "still shopping around" implies that a suitable option has yet to be found. What's more, the dataset also contains entries that (again, in our judgment) simply do not contain implicatures. In the following example, the response appears to be a direct answer to the question (even though it does not contain the word 'yes' or 'no').

> "Did he ever fall back on a run?"
> "All the time, sir." (Sorkin, 1991)

These patterns show a potential issue in using the George and Mamidi (2020) dataset to evaluate models' performance at recovering implicatures. The BIG-bench implicature task uses the same dataset, but narrows it down to a greater extent—such as by "[d]iscarding factual errors in the original dataset" (Maru and Bevilacqua, 2022). This further constrained dataset may be useful in accurately identifying models' performance at implicature recovery, but of course comes at the expense of being even smaller. Additionally, there are a number of other datasets that could be used to similarly evaluate models' performance, however they are not without their own pitfalls.

The GRICE dataset is a collection of conversations involving implicatures and multiple-choice style questions, the correct answers to which depend on recovering the implicature (Zheng et al., 2021). Unlike the George and Mamidi (2020) dataset, Zheng et al. do not explicitly annotate the

Prompt	k	Implicature	Completion
Template 1	0	0.8%	2.5%
	1	62.3%	83.0%
Template 4	0	1.2%	1.7%
	1	65.2%	87.2%
Template 5	0	1.5%	2.3%
	1	66.0%	90.8%
Modified Template 1	0	79.3%	100.0%
	1	77.7%	100.0%
Modified Template 4	0	78.8%	100.0%
	1	78.0%	100.0%
Modified Template 5	0	80.3%	100.0%
	1	78.8%	100.0%

Table 2: Breakdown of Cohere Command R+ implicature and completion accuracy across the original "Finish the following text:" prompts from Ruis et al. (2024) and our modified prompts.

value of the implicature in each conversation, but instead only which of the multiple choice answers is correct. The GRICE dataset could be used in conjunction with the likelihood based approach used in Ruis et al. (2024) (see background in Section 2.2) by evaluating which of the multiple-choice answers the model predicts is most likely to appear. Because the data is programatically generated, however, this may exhibit the same issue of unnaturalness as in George and Mamidi (2020). In that regard, the variety of the GRICE data is rather limited: there are only four subtopics used to generate the conversations, which all follow a relatively simple conversational structure.

The dataset used in de Marneffe et al. (2010) provides a more natural source of implicature data. The authors sourced data from transcripts of interviews aired on CNN from 2000–2008 and the Switchboard corpus of telephone conversations (see Jurafsky et al., 1997). Labels were assigned based on the distribution of judgments of 30 Mechanical Turk workers for each of the dialogues. This may provide a higher quality source of data for evaluating implicature recovery performance, but it comes at the expense of being substantially smaller ($n = 224$).

One of the larger extant datasets is the Circa dataset, comprising 34,000+ pairs of crowdsourced questions and indirect answers (Louis et al., 2020). Both the questions and answers are crowdsourced. Labeling of the answers is also crowdsourced and divides the answers into yes/no categories (along with a split between certain/strong and uncertain/weak) as well as unsure and 'in the middle' (neither yes nor no) categories. The Louis et al. dataset seems promising as it is substantially larger than any of the others considered.

While the particular examples we discuss are specific to conversational implicature, they are illustrative of the potential issues that can arise when relying uncritically on existing datasets or benchmarks and using them to evaluate different models. The nature and quality of a particular dataset can play a significant role in a model's performance, and can risk presenting a distorted picture when attempting to make comparisons across models (let alone across datasets/benchmarks).

4.2 Prompt Sensitivity

Next, we turn to the issue of prompt sensitivity when it comes to cross-model comparisons and structured generation as a potential solution. Our experiment contributes further evidence to discussion in the literature regarding the danger of conceptualizing prompting as just another way of getting answers from a machine, comparable to the algorithms of prior generations. For example, Loya et al., 2023 found that GPT-3.5-Turbo's performance on a task conducted in prior research could be worsened or significantly improved with relatively minor alterations to the prompt. Our results in Section 3.3 reinforce the point: a difference of just four words ("with yes or no") dramatically changed the model's score on this benchmark. These observations suggest that the sensitivity of performance to prompt specifics is an essential consideration in any experiment using LLMs, and tools for evaluating prompt sensitivity (e.g., Sclar et al., 2024; Zhuo et al., 2024) should be a part of any future benchmark development process.

In terms of mitigating the risks of prompt sensitivity, Ruis et al. (2024) did so, to some extent, by using a set of six different prompts, rather than a single one. They divide the prompts into two groups: natural (prompts 1, 4, and 5) and structured (prompts 0, 2, and 3). However, as shown by the results with Command R+ (see Section 3.3), this was not entirely successful: Command R+ has consistent performance across prompts within a single group, but performs substantially differently between the natural prompts and the structured prompts.

In addition, chain-of-thought prompting (Wei et al., 2022), one of the techniques used by Loya et al., is also explored in Ruis et al. (2024). They found that 5-shot evaluation with chain-of-thought prompting brought GPT-4 to comparable performance to their human baseline. This improvement over the non-chain-of-thought results suggests that it is difficult—through completion tasks alone—to determine to what extent a language model has captured generalizations about implicatures.

Another way of avoiding the inherent prompt sensitivity of large language models is to avoid using text-generation tasks to study them. Instead, Ruis et al.'s (2024) comparing the relative likelihoods of multiple possible options would be more resilient to minor variations in the prompt. Unfortunately, the fact that state-of-the-art language models are developed by corporations that do not publish the full models presents a roadblock to studying them in more detail (e.g., OpenAI, 2023b, "Given both the competitive landscape and the safety implications of large-scale models like GPT-4, this report contains no further details."). Because access to the models is gated behind corporate APIs, which do not provide this information, research like ours is unable to use this technique.

Before we turn to structured output as a potential method for addressing the pitfalls we found associated with prompt sensitivity, we again emphasize that, although the specific issues discussed here depend on the use case and particular models under consideration, the broader issue of prompt sensitivity is fundamental to all large language models, including both closed source and open source (Sclar et al., 2024). As Errica et al. (2025) note, results from any model trained to maximize a likelihood objective are going to be sensitive to all features of the prompt that affect its probability.

4.3 Structured Output

In the interval between our original experimentation and writing this paper, structured output became an option for many LLMs: it is possible to make LLM text-generation requests explicitly defining the desired output format and limiting the model's output to that which conforms to the specified format. OpenAI's API now supports structured output by allowing the user to provide a JSON schema which the output must match (Pokrass, 2024): they describe a sampling process during text generation as "determin[ing] which tokens are valid to be produced next based on the previously generated tokens and the rules within the grammar that indicate which tokens are valid next." Ollama similarly supports providing a JSON schema to restrict the output (Ollama, 2024). Perhaps this renders many prompt sensitivity concerns moot?

We tested both GPT-4o and Llama 3.2 using a version of the Ruis et al. (2024) task adapted to use structured output. Rather than directly parsing the text, we used a JSON schema to have the model generate a JSON object containing a single boolean property representing the value of the implicature.

It turns out that, although structured output helps, LLMs persist in being inappropriately sensitive to details of the way they are called. In particular, note that in defining the JSON schema for the output, we were faced with the choice of what *name* to give to the boolean property representing the recovered value of the implicature. Although initially the grammar constrains the possible tokens to produce the JSON key, notice that, per the quote above, the key itself is part of the context and thus *the name of the key* will affect how the value for that property is generated.

To confirm this makes a difference, we tested both GPT-4o and Llama 3.2 using the original prompt templates from Ruis et al. (2024) using several different names for the boolean property, the results of which are shown in Table 3. We found that the property name can have a significant impact, though to what extent is variable.

Furthermore, we found that performance is still somewhat sensitive to the prompt, despite the constraints on the output. Table 4 shows the accuracy of Llama 3.2 for each prompt template in the structured output task. We note that adding "with yes or no" to prompt templates 1, 4, and 5 still produces a marked accuracy difference. That said, we also note that unmodified templates (0, 2, and 3) exhibit

JSON Key	GPT-4o	Llama 3.2
answer_is_yes	80.3% ± 6.0	60.2% ± 4.7
implicature_is_yes	80.2% ± 5.5	56.5% ± 2.5
implicature_value	70.2% ± 14.4	55.2% ± 3.1

Table 3: Mean accuracy across prompt templates for GPT-4o and Llama 3.2 depending on the key name in the JSON schema, when tested with the unmodified prompt templates and $k = 0$.

Prompt	Original	Modified
Template 0	64.0%	63.0%
Template 1	55.7%	60.8%
Template 2	65.0%	61.8%
Template 3	65.5%	65.0%
Template 4	55.0%	58.7%
Template 5	56.0%	61.2%

Table 4: Structured output accuracy for Llama 3.2 across the original and modified prompt templates (for templates 1, 4, and 5) when tested with answer_is_yes as the JSON key for $k = 0$.

a similar difference in some cases, so this effect may be within the run-to-run variance of the test.

Overall, we find that, although structured output may address the challenge of extracting information from LLM output, prompt sensitivity remains a significant concern. Put plainly: structured output affects the output's *structure*, not its substantive *content*. Instructions given to the model continue to have an impact on its apparent performance at a task, even if the model now always produces "grammatically correct" output. Additionally, structured output introduces the additional challenge of the output grammar itself (such as the names of the JSON keys) also affecting performance.

4.4 Other Paths Forward

As an alternative to seeking LLM-engineering solutions to the problems we are describing—something that in our view requires the efforts of the entire broader community—we conclude our discussion by considering underlying properties of the linguistic phenomenon being studied as a potentially more effective way to analyze the capabilities of language models. This can be thought of as a general strategy that we apply here to the specifics of conversational implicatures as a problem space.

Defeasibility and Reinforceability of Implicatures Two of the characteristic features of implicatures are that they are both defeasible and reinforceable (Levinson, 1983). They are defeasible in that the speaker of an implication-carrying utterance can defeat or cancel the implication in a subsequent utterance (for example, by saying something along the lines of, "But it's not actually the case that <*implication*>."). Similarly, they are reinforceable, and the speaker could emphasize what was previously implied. It's important to note that what makes the case of an implicature different from another utterance is that defeating or reinforcing an implication-carrying utterance neither produces a contradiction nor sounds redundant. By contrast, attempting to defeat an ordinary sentence does result in a contradiction and attempting to reinforce it often sounds redundant.[4]

Those differences could be used to test a model's sensitivity to implicature in a context where the likelihood of a string can be obtained from the model. By starting with a single question and an answer to it phrased both explicitly and as an implicature, and then comparing the likelihood of each of those being followed by a sentence that defeats/contradicts it, it may be possible to identify whether the model has recovered the implicature and the fact that it is an implicature. Flatly contradicting a prior sentence should be relatively unlikely. But, if the model has identified the implicature, then defeating it should be substantially more likely than the case of contradiction. Similarly, a sentence that repeats the same meaning as the previous one should be less likely in the case where the previous sentence is explicitly saying the same thing as compared to when the meaning of the previous sentence is provided by implication.

Unfortunately, this hypothesis is not readily testable at present, owing to the lack of likelihood information provided by the APIs for state-of-the-art language models.

Direct Inquiry vs. Conversation Continuations Our final observation is that evaluating language models' competence at recovering implicatures using a strategy of simply prompting them with instructions to evaluate the yes/no value of an implicature may not effectively represent their use of implicature in conversations. Presumably little of the models' training datasets consists of people directly asking what the meaning of an implication-carrying sentence is (aside, perhaps, from students of semantics or pragmatics). It is more likely that

[4] Levinson (1983) notes that there are circumstances, such as involving stress, where other types of sentences can be reinforced without issue. But those are not germane to our discussion.

the use of implicatures in the wild—and the conversations flowing therefrom—are better represented in the training data.

Since large language models are fundamentally constructed as text prediction/generation systems (e.g., "GPT-4 is a Transformer-style model pretrained to predict the next token in a document" OpenAI, 2023b), a task aimed at probing the same question but formulated to the context of text prediction/generation may produce more representative results. For example, given a context question and a response utterance carrying a conversational implicature, using a language model to generate a continuation of that conversation may provide another avenue for determining whether the model recovered the value of the implicature. If the model has recovered the value of the implicature, then the generated conversation should continue to flow naturally. If it has not, then there would be a break in the common ground and the conversation should be anomalous in some way.

5 Conclusions

With regard to conversational implicature, we have contributed an updated evaluation showing that Ruis et al.'s (2024) results hold up, improve with newer models, and that hoped-for improvements when moving from 0-shot to 1-shot to 5-shot in-context learning are not consistent. In addition, however, our simple attempt at replicating prior work using more up-to-date LLMs foregrounded deeper issues, ones that connect to broader questions about how to use and evaluate LLMs.

One key takeaway involves *data quality*, which receives little attention in NLP. In contrast to other fields like survey research and social sciences that have developed established, systematic frameworks for data quality assessment (Pipino et al., 2002; Groves and Lyberg, 2010; Birkenmaier et al., 2024), NLP research still largely lacks such frameworks and, despite some recognition of the problems (Bender and Friedman, 2018; Gebru et al., 2021; Northcutt et al., 2021) and emerging efforts to systematize data quality approaches (Dang and Verma, 2024; Mishra et al., 2020), there is scant evidence to suggest that common best practices are moving in that direction.

A second takeaway concerns the use of completion-based tasks. Our results and discussion suggest that completion-based tasks should be viewed with greater caution than they presently are, particularly for reasons associated with prompt sensitivity. Unfortunately, the constraints commercial LLM providers place on availability for alternatives, e.g. use of likelihoods, stymie otherwise potentially useful and creative solutions. We have suggested that in the absence of general solutions, finding ways to exploit relevant properties of the problem may be a better, or at least complementary, path forward.

A third takeaway concerns the pace of change in NLP. We attempted replication because models are constantly being updated. Having identified a problem with insufficiently constrained LLM output, we introduced solutions (e.g. prompt rephrasing)—only to find that by the time we were writing about the effort, still more recent developments in structured output capabilities required their own experimentation and evaluation, *and*, naturally, still did not fully fix the problem. Our takeway here is that the remarkably rapid change in NLP is both a blessing and a curse: in general we obtain better and better models and approaches, but there is barely any time to actually think deeply when so much effort is needed just to keep up. We would suggest that the field could benefit from a dose of slow science (Stengers, 2018), a perspective that de-emphasizes performance targets, deadlines, and market-based influences in favor of deeper thinking and curiosity-driven progress.

Finally, it is worth considering here, as with any attempt at creating an objective benchmark to measure the quality of a large language model, how the metric being used relates to the actual goal being pursued. Achieving a perfect score—or even a human-level score, like GPT-4—does not mean that a model has necessarily captured the same generalizations about implicatures that humans have. It may be that building or refining a model in order to improve its score on the Ruis et al. (2024) benchmark is not necessarily a productive way of improving its actual ability to *use* implicature. The broader take-away message is that we would do well to reminder ourselves regularly that "when a measure becomes a target, it ceases to be a good measure" (Goodhart's Law, Strathern, 1997).

Acknowledgments We are grateful to our anonymous reviewers for their thoughtful comments that improved the quality of the paper, Ruis et al. for their code and data, and the providers of the models that we used in our experimentation.

References

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

Lena Birkenmaier, Jessica Daikeler, Lisa Fröhling, Tobias Gummer, Clemens Lechner, Vanessa Lux, and Sebastian Ziaja. 2024. Defining and evaluating data quality for the social sciences: Position paper. GESIS Papers, 2024/06. Köln: GESIS – Leibniz-Institut für Sozialwissenschaften.

Viet Minh Hoang Dang and Rakesh M Verma. 2024. Data quality in nlp: Metrics and a comprehensive taxonomy. In *Advances in Intelligent Data Analysis XXII*, volume 14641 of *Lecture Notes in Computer Science*, pages 305–318. Springer.

Wayne Davis. 2024. Implicature. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*, Spring 2024 edition. Metaphysics Research Lab, Stanford University.

Marie-Catherine de Marneffe, Christopher D. Manning, and Christopher Potts. 2010. "Was it good? It was provocative." Learning the meaning of scalar adjectives. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 167–176, Uppsala, Sweden. Association for Computational Linguistics.

English Test Store. `https://englishteststore.net/`. Accessed: 2025-06-02. [link].

Federico Errica, Davide Sanvito, Giuseppe Siracusano, and Roberto Bifulco. 2025. What did I do wrong? quantifying LLMs' sensitivity and consistency to prompt engineering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1543–1558, Albuquerque, New Mexico. Association for Computational Linguistics.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92.

Elizabeth Jasmi George and Radhika Mamidi. 2020. Conversational implicatures in english dialogue: Annotated dataset. *Procedia Computer Science*, 171:2316–2323. Third International Conference on Computing and Network Communications (CoCoNet'19).

Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Robert M Groves and Lars Lyberg. 2010. Total survey error: Past, present, and future. *Public Opinion Quarterly*, 74(5):849–879.

Alexander Hoyle, Pranav Goel, Andrew Hian-Cheong, Denis Peskov, Jordan Boyd-Graber, and Philip Resnik. 2021. Is automated topic model evaluation broken? the incoherence of coherence. *Advances in neural information processing systems*, 34:2018–2033.

Jennifer Hu, Sammy Floyd, Olessia Jouravlev, Evelina Fedorenko, and Edward Gibson. 2023. A fine-grained comparison of pragmatic language understanding in humans and language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4194–4213, Toronto, Canada. Association for Computational Linguistics.

IMSDb. The internet movie script database. `https://imsdb.com/`. Accessed: 2025-06-02.

Daniel Jurafsky, Elizabeth Shriberg, and Debra Biasca. 1997. Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13. Technical Report 97-02, University of Colorado, Boulder Institute of Cognitive Science, Boulder, CO.

Stephen C. Levinson. 1983. Conversational implicature. In *Pragmatics*, chapter 3. Cambridge University Press.

Annie Louis, Dan Roth, and Filip Radlinski. 2020. "I'd rather just go to bed": Understanding indirect answers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7411–7425, Online. Association for Computational Linguistics.

Manikanta Loya, Divya Sinha, and Richard Futrell. 2023. Exploring the sensitivity of LLMs' decision-making capabilities: Insights from prompt variations and hyperparameters. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3711–3716, Singapore. Association for Computational Linguistics.

Marco Maru and Michele Bevilacqua. 2022. Implicatures. README.md.

Swaroop Mishra, Anjana Arunkumar, Bhavdeep Singh Sachdeva, Chris Bryan, and Chitta Baral. 2020. Dqi: Measuring data quality in nlp. *arXiv preprint arXiv:2005.00816*. Available at: `https://arxiv.org/abs/2005.00816`.

Curtis Northcutt, Anish Athalye, and Jonas Mueller. 2021. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Ollama. 2024. Structured outputs. `https://ollama.com/blog/structured-outputs`. Accessed: 2025-06-02.

OpenAI. Models. `https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4`. Accessed: 2025-06-02.

OpenAI. 2023a. GPT-4 API general availability and deprecation of older models in the completions API. https://openai.com/index/gpt-4-api-general-availability/. Accessed: 2025-06-02.

OpenAI. 2023b. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

Leo L Pipino, Yang W Lee, and Richard Y Wang. 2002. Data quality assessment. *Communications of the ACM*, 45(4):211–218.

Michelle Pokrass. 2024. Introducing structured outputs in the api. https://openai.com/index/introducing-structured-outputs-in-the-api/. Accessed: 2025-06-02.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Deborah Rossen-Knill, Beverly Spejewski, Beth Ann Hockey, Stephen Isard, and Matthew Stone. 1997. Yes/no questions and answers in the map task corpus.

Laura Ruis, Akbir Khan, Stella Biderman, Sara Hooker, Tim Rocktäschel, and Edward Grefenstette. 2024. The goldilocks of pragmatic understanding: fine-tuning strategy matters for implicature resolution by llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, et al. 2024. The prompt report: A systematic survey of prompting techniques. *arXiv preprint arXiv:2406.06608*. Available at: https://arxiv.org/abs/2406.06608.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *Preprint*, arXiv:2310.11324.

Aaron Sorkin. 1991. A few good men. https://imsdb.com/scripts/A-Few-Good-Men.html. Accessed: 2025-06-02.

Dan Sperber and Deirdre Wilson. 1986. *Relevance: Communication and Cognition*. Harvard University Press.

Neha Srikanth, Rupak Sarkar, Heran Mane, Elizabeth Aparicio, Quynh Nguyen, Rachel Rudinger, and Jordan Boyd-Graber. 2024. Pregnant questions: The importance of pragmatic awareness in maternal health question answering. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7253–7268, Mexico City, Mexico. Association for Computational Linguistics.

Isabelle Stengers. 2018. Another science is possible: A manifesto for slow science transl. *S Mueke Cambridge, UK: Polity (from French)*.

Marilyn Strathern. 1997. 'Improving ratings': audit in the British university system. *European Review*, 5(3):305–321.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zilong Zheng, Shuwen Qiu, Lifeng Fan, Yixin Zhu, and Song-Chun Zhu. 2021. GRICE: A grammar-based dataset for recovering implicature and conversational reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2074–2085, Online. Association for Computational Linguistics.

Jingming Zhuo, Songyang Zhang, Xinyu Fang, Haodong Duan, Dahua Lin, and Kai Chen. 2024. ProSA: Assessing and understanding the prompt sensitivity of LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1950–1976, Miami, Florida, USA. Association for Computational Linguistics.

A Prompt Templates

Table 5 shows the original prompts from Ruis et al. (2024). Those that were modified in our experiment were changed to read "Finish the following text with yes or no:" for the instruction.

#	Original template	Modified in our experiment
0	Does the following response to the question imply yes or no? question: <*context utterance*> response: <*response utterance*> implicature:	No
1	Finish the following text: Esther asked "<*context utterance*>" and Juan responded "<*response utterance*>", which means	Yes
2	Is the implied meaning of the following response yes or no: question: <*context utterance*> response: <*response utterance*> meaning:	No
3	What is the intent of the following response, yes or no? question: <*context utterance*> response: <*response utterance*> intent:	No
4	Finish the following text: Karen asked "<*context utterance*>" and William responded "<*response utterance*>", which means	Yes
5	Finish the following text: Bob asked "<*context utterance*>" and Alice responded "<*response utterance*>", which means	Yes

Table 5: The prompt templates from Ruis et al. (2024) and whether they were modified in our experiment.

Are syntactic categories ISL-2 inferrable? A corpus study

Logan Swanson and **Kenneth Hanson** and **Thomas Graf**
Stony Brook University
logan.swanson@stonybrook.edu
mail@kennethhanson.net
mail@thomasgraf.net

Abstract

We use the MG treebank of Torr (2017) to investigate the conjecture in Graf (2020) that category systems are ISL-2 inferrable. A category system is ISL-2 inferrable iff the category feature of every lexical item can be jointly inferred from phonological exponents of both the item itself and either its selecting head or the arguments it selects. If correct, this conjecture would greatly limit the overgeneration problem posed by subcategorization mechanisms (Kobele, 2011; Graf, 2011, 2017). We find that the conjecture is largely borne out in this data set. However, we also observe that it holds even for features that are not expected to be inferrable in this manner, and we demonstrate that inferrability can arise from the assumption that certain distributional properties of the lexicon are Zipfian in nature. We conclude that category systems in natural languages may well be ISL-2 inferrable, but that this could be due to extragrammatical factors.

1 Introduction

A good model of language should be sufficiently expressive to account for observed linguistic variation while still being restrictive enough to rule out highly unnatural patterns. Graf (2017) highlights a major overgeneration problem with syntactic *subcategorization* mechanisms. Subcategorization is needed to capture basic facts such as *devour* being a verb that takes a DP subject and a DP object. But without meaningful restrictions on the inventory of syntactic categories, subcategorization can be used to enforce *any* constraint definable in monadic second-order logic (MSO).

MSO has been used extensively in model-theoretic syntax (see Rogers 1998, Rogers 2003, Morawietz 2003, Tiede and Kepser 2009, Graf 2013, and references therein) due to its ability to succinctly capture even the most byzantine proposals from the syntactic literature. However, it can also express highly unnatural constraints such as "a reflexive must c-command a verb of motion unless there are at least three CP nodes in the same tree that each properly dominate an odd number of nodes". Extending a well-known translation mechanism from MSO constraints to bottom-up tree automata (Thatcher and Wright, 1968; Doner, 1970), the states of these automata can be compiled into a grammar's category system to implicitly enforce MSO constraints via subcategorization (Graf, 2011; Kobele, 2011). Graf (2017) argues that linguists' restrictions on category systems are not tight enough to rein in subcategorization, and as a result current theories of syntax are much less restrictive than they appear.

Graf (2020) shows that many undesirable kinds of overgeneration, e.g. modulo counting, can be ruled out if category features are required to be inferrable by *input strictly 2-local* (ISL-2) functions. Intuitively, the category feature of a lexical item *l* is *ISL-2 inferrable* iff it can be predicted from the phonological content of *l* itself and its local tree context. Graf (2020) conjectures that all natural languages have category systems that are ISL-2 inferrable. If true, this would explain how subcategorization can be ubiquitous in syntax without giving rise to unnatural MSO constraints.

In order to assess the viability of ISL-2 inferrability as a linguistic universal, we test whether it holds for MGBank (Torr, 2017), a treebank of English sentences with structures very similar to those assumed by Graf (2020). We find that the category features for a large majority of lexical items can indeed be predicted from strictly local tree contexts. When a category feature is not ISL-2 inferrable, that is usually due to empty heads, i.e. lexical items that lack phonological exponents and hence provide no information for ISL-2 inferrability (an edge case already mentioned in Graf 2020). However, we also find a similarly high degree of inferrability for movement features, which operate over long distances and would not be expected to be ISL inferrable by this conjecture. Probing further, we show that ISL-2 inferrability can arise in language datasets following Zipfian frequency distributions. This makes it difficult to assess whether ISL-2 inferrability is a guiding principle of the grammar, as conjectured by Graf (2017), or rather an artifact of other features of human language.

This paper is organized as follows. Section 2 introduces the necessary background on the Minimalist grammar formalism (Sec. 2.1), the overgeneration problem (Sec. 2.2), and ISL-2 inferrability (Sec. 2.3). Section 3 describes the data and methodology used. Section 4 displays our findings on the ISL-inferrability of category-system features and discusses how they may support the ISL-inferrability hypothesis. Section 5 complicates this picture by introducing theoretical limitations of ISL-2 inferrability and also demonstrates how a high degree of inferrability can arise naturally from other properties of language. Section 6 offers ideas for future research directions and concludes.

2 Background: ISL inferrability

2.1 Categories in Minimalist grammars

Following Graf (2017), the results of this paper are couched in the formal terms of *Minimalist Grammars* (MG) (Stabler, 1997, 2011) and *suregular syntax* (Graf, 2022b,a). However, the results of this study are not limited in relevance to just those formalisms and bear on syntax much more generally. ISL-2 inferrability asks whether certain kinds of information can be inferred from local tree contexts, and in MG trees the local relationships are those between heads and their arguments (specifiers and complements). The central question that Graf (2017) formally hashes out as ISL-2 inferrability over MG trees thus is much broader and extends far beyond MGs to other formalisms: to what extent can specific features of a lexical item be inferred from the phonological content of its arguments and/or its selecting head?

In MGs, every lexical item consists of a *phonological exponent* that determines its pronunciation, and a string of features that determine its syntactic behavior. The feature string always contains a *category feature* (x) and may contain *selector features* (=x) that encode the item's subcategorization requirements. For example, a word like *say* would have the feature string ⟨=c =d v⟩, representing that it selects a CP complement, a DP specifier, and is a verb.

The MG feature strings may also include movement features. The *licensee feature* -m indicates that the item is a mover of type m, while a *licensor feature* +m indicates that this item furnishes a landing site that must be filled by an m-mover. Graf (2020) explicitly states that movement features are not expected to be ISL-2 inferrable. This effectively makes inferrability of movement features a "control group" for our corpus experiment, a point we will return to in Section 5.2. Until then, we omit movement features from the discussion and all examples.

MGs furnish multiple types of structural descriptions: phrase structure trees, derivation trees, and dependency trees. While a lot of early MG work focused on phrase structure trees, Kobele et al. (2007) started a shift toward derivation trees as the primary syntactic representation of MGs. Derivation trees are also used in the MG treebank (Torr, 2017) that our corpus analysis is based on. Subregular syntax, including Graf (2017), prefers dependency trees instead. But since there is a one-to-one correspondence between derivation trees and dependency trees, the choice is purely a matter of mathematical convenience and it is easy to translate between the two.[1] Graf (2017) uses dependency trees because of their close connection to head-argument relations: the mother-of relation in MG dependency trees encodes subcategorization. Every node is a (feature-annotated) lexical item, and its *i*-th daughter from the right is its *i*-th argument — the rightmost daughter is the complement, all other daughters are specifiers. Even though MGs use movement, no displacement takes place in dependency trees. Every lexical item sits in the position where it is selected, and movement is encoded purely via movement features. An example tree for the sentence *The child laughed at a bear* is given in Fig. 1.

2.2 The overgeneration problem

Although subcategorization is crucial for modeling the kinds of patterns found in syntax, it introduces

[1] As pointed out in Graf (2011, 2012), MG derivation trees are built from chunks of derivational structure called *slices*. Intuitively, the slice slice(l) consists of the operations that assemble the projections of lexical item l in the phrase structure tree. A given MG derivation tree t is converted to an equivalent MG dependency tree by replacing slice(l) with l for every lexical item l of t. For example, if slice(l) = Move(Merge(x, Merge(l, y))), this is condensed to $l(x,y)$. One could also say that MG dependency trees are the derivation trees of a Tree Substitution Grammar that generates MG derivation trees.

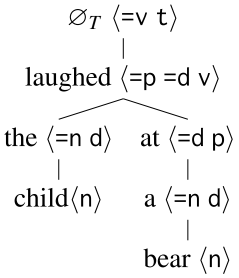


Figure 1: MG dependency tree for *The child laughed at a bear*, with empty T-head above the verb

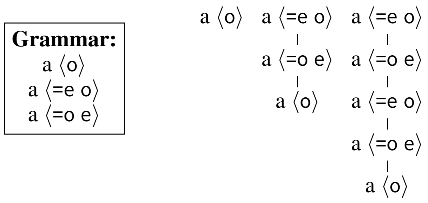


Figure 2: Smuggling in an unnatural *modulo* counting constraint via the category system. Left: Grammar which tracks o[dd] and e[ven] nodes, Right: Some trees generated by this grammar.

massive overgeneration into the formalism. As mentioned in the introduction, Graf (2011) and Kobele (2011) show that a constraint can be enforced via MG-style subcategorization iff it is definable in MSO. Figure 2 gives an example where the category system is used to track whether a subtree contains an odd (o) or an even (e) number of nodes. Graf (2017) illustrates the many ways MSO-constraints and, by extension, subcategorization undermine the restrictiveness of syntactic formalisms. A restrictive theory of syntax thus requires tight restrictions on its category system.

2.3 ISL-2 inferrability to the rescue

Graf (2020) proposes to curb the excessive power of subcategorization by requiring category features to be inferrable by input strictly 2-local (ISL-2) tree-to-tree transductions. While the definition of ISL-2 transductions in Graf (2020) is fairly technical, the general idea is simple enough (see Fig. 3 for a visualization).

Suppose we take a dependency tree t generated by some MG G and remove all feature strings from all nodes, leaving only the exponents. Is there a function f_G that correctly determines for each node n of t whether n had feature f? If the answer is positive for every node of every dependency tree of G, then f is *inferrable* for G. If f_G can do this based solely on I) the exponent of n and II) the exponents of either IIA) n's mother and siblings (*upper context*) or IIB) the exponents of n's daughters (*lower context*), then f is *ISL-2 inferrable* for G.

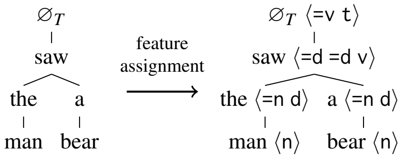


Figure 3: Feature assignment transduction

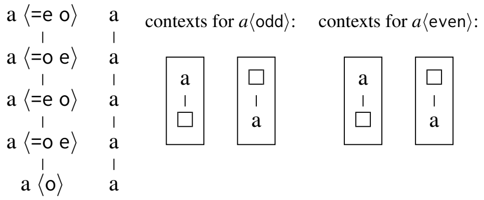


Figure 4: Category system implementing modulo counting is not ISL inferrable

Many unnatural category systems, like the *modulo* counting example in Fig. 2, are not ISL-2 inferrable. Figure 4 shows that the category features o and e are not ISL-2 inferrable because they share at least one structural context of size 2 (in fact, their contexts are exactly the same). Meanwhile, many natural patterns which require subcategorization *are* ISL-2 inferrable: Fig. 5 demonstrates how local contexts can successfully disambiguate two lexical entries for *have*. In light of this, Graf (2020) conjectures that ISL-2 inferrability (or at least ISL-k inferrability for some fixed $k \geq 2$) is a linguistic universal of category systems. Next, we will evaluate this conjecture with our corpus study.

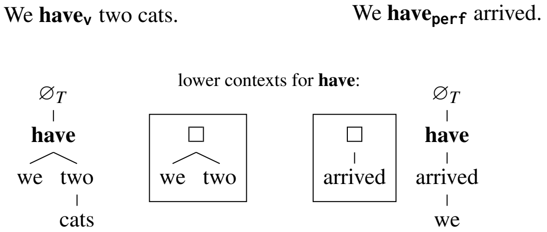


Figure 5: Example of disambiguating contexts for two lexical entries of *have*

3 Methods

3.1 Corpus: MGBank

To investigate the viability of ISL-2 inferrability as a linguistic universal, we conducted a study using data from MGBank (Torr, 2017), a database of MG derivation trees. The data in MGBank was created by automatically translating a portion of the Penn Treebank (Marcus et al., 1993) followed by a manual check for correctness. Overall, MGBank consists of 49,000 Wall Street Journal sentences, adding up to over 1 million words. While the derivation tree format in MGBank is different from the dependency tree format used here, there is a deterministic, sound, and complete translation from the former to the latter (see fn. 1). These qualities make MGBank an ideal data set for testing the conjecture that syntactic categories are ISL-2 inferrable.

3.2 Determining ISL-2 inferrability

Data from MGBank was first converted from MG derivation trees into dependency trees.[2] The MGBank annotation scheme includes some details which are not relevant to our research question and were therefore removed as part of the translation. For example, information on whether an argument should be linearized to the left or the right of the head was removed. Additionally, adjunction was converted to category-preserving selection with empty heads (the consequences of adjunction are discussed in Section 5.1). Next, a lexicon was extracted, consisting of all attested pairs of exponents and feature strings.

We then examined inferrability for category features in isolation as well as category features together with selector features. From a linguistic perspective, the category features are more important, since once these are determined, the selector features follow trivially. As a control, we also test inferrability for movement features.

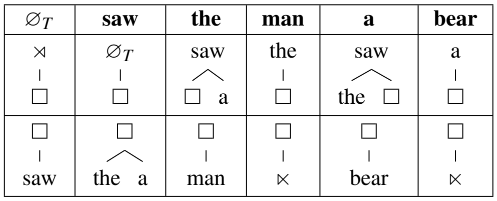


Figure 6: Upper and lower (size 2) contexts for each lexical item in the sentence *The man saw a bear*.

Lexical Item	Contexts	Unique	Shared
a ⟨fspec1⟩	{ c1,c2,c3 }	{ c1,c2,c3 }	{ }
a ⟨fspec2⟩	{ c4,c5,c6 }	{ c4,c5,c6 }	{ }
b ⟨fspec1⟩	{ c1,c2,c3 }	{ c1 }	{ c2,c3 }
b ⟨fspec2⟩	{ c2,c3,c4 }	{ c4 }	{ c2,c3 }
c ⟨fspec1⟩	{ c1,c2,c3 }	{ }	{ c1,c2,c3 }
c ⟨fspec2⟩	{ c1,c2,c3 }	{ }	{ c1,c2,c3 }

Figure 7: Computing shared and unique contexts for each lexical item. The features for items with exponent *a* are strongly (and also weakly) inferrable, those for items with exponent *b* are weakly (but *not* strongly) inferrable, and those for items with exponent *c* are neither.

In many cases, the relevant features (category / category + selector / movement) are predictable directly from the exponent itself. This means that they are ISL-1 inferrable and hence ISL-2 inferrable. For example, the category of *destruction* is always n irrespective of its local context. ISL-1 inferrability of feature f can fail only if the corpus contains two lexical items l and l' such that both have the same exponent but only of them carries f. But f can still be ISL-2 inferrable if l and l' have distinct structural contexts.

Given a node n in tree t, its *upper context* consists of n itself, its parent, and any siblings of n, while the *lower context* consists of n itself and its children. Crucially, our contexts track only exponents, with all features omitted. Following Graf (2020), we modified each tree by inserting ⋊ above the root and ⋉ below each leaf so that every lexical item has an upper and a lower context in every tree. Figure 6 gives an example for the upper and lower contexts for each element in the example from Fig. 3.

The following method is used to assess ISL-2 inferrability of a given feature (or string of features) f: First, the set of all lexical items is extracted from the corpus together with the upper and lower contexts for each lexical item. This then allows us to assess two types of ISL-inferrability in terms of context sets. For each exponent e, let E be the set of lexical items that share the same exponent. We say that f is *strongly inferrable* iff it holds for every exponent e that no $l \in E$ carrying f ever appears in the same (upper or lower) context as some $l' \in E$ without f. We also say that l and l' have no *shared* contexts. When the contexts are restricted to upper and lower contexts as defined above, f being strongly inferrable is equivalent to it being ISL-2 inferrable. We say that f is *weakly inferrable* iff it

[2] Complete code for this project can be found at www.github.com/pterodactylogan/isl-k-corpus-test

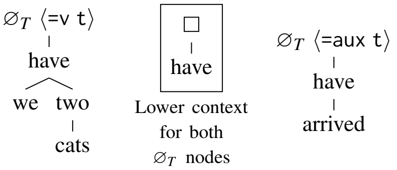


Figure 8: Inferrability is difficult with empty heads. Here, the lower context is insufficient to discriminate between the T head which selects a v complement (left) and the one which selects a aux complement (right).

holds for every exponent e and every $l \in E$ carrying f that l occurs in some (upper or lower) context that no $l' \in E$ without f occurs in. We also say that l has a *unique* context. While weak inferrability does not imply ISL-2 inferrability, it was included in this study because it might be a useful property for distributional learning algorithms. Weak and strong inferrability of each feature were then computed for each lexical item using these contexts. Figure 7 illustrates this process.

3.3 The trouble with empty heads

A possible stumbling block for ISL-inferrability comes from empty heads, which have no pronounced exponent. Empty heads introduce a lot of ambiguity, particularly when many of them are stacked together, e.g. in the functional hierarchy C-T-*v*-V commonly assumed in Minimalism. Figure 8 illustrates this issue with an empty T-head.

At the same time, these heads may actually carry prosodic information (e.g. a C-head that furnishes a wh-landing site) or contribute information that is pronounced on other heads, like tense. Arguably, this information should be taken into account for ISL-2 inferrability. In the following section, we report results with this information (empty heads have exponents such as [PAST] or [PRESENT]) and without (empty heads have the empty string as their exponent).

4 Results

4.1 Strong support for ISL-2 inferrability

We now report our findings on the inferrability of feature strings in MGBank. The full corpus contains nearly 40,000 distinct lexical items, with each lexical item including an exponent, a category feature, and zero or more selector and movement features. As mentioned above, we examined various subsets of features, and tested inferrability both with and without disambiguation of empty heads. Both of these variables affect the total number of distinct items, which we report along with results on inferrability.

For each of the feature subsets discussed in Section 3.2, the total number of *ambiguous* items was computed, that is, those that are *not* inferrable. This was done based on the criterion for ISL-1 inferrability as well as both strong and weak ISL-2 inferrability. The level of ISL-1 inferrability reflects the amount of *lexical ambiguity* in the corpus. The percentages for both weak and strong ISL-2 inferrability therefore indicate the percentage of lexically ambiguous items (rather than all items) which cannot be disambiguated using a context of size 2. Because the number of lexically ambiguous items may be much smaller than the total, taking the latter as a baseline could create a skewed view of how much work the local structural context does to disambiguate category information.

Table 1 shows the inferrability for category features and category + selector features depending on whether empty heads have as their exponent the empty string or linguistic annotations like [PAST]. These results demonstrate that ISL-inferrability holds for the vast majority of lexical items (*modulo* movement features). In fact, nearly two-thirds of category features and nearly half of category + selector feature pairs are ISL-1 inferrable. Category features alone have much less ISL-1 (lexical) ambiguity than category + selector features together, which is unsurprising as it is common for a word to correspond to multiple lexical items that differ in their subcategorization properties but still have the same category. Interestingly, more of this ambiguity can be resolved by ISL-2 contexts with category + selector pairs than category features alone. Overall, category feature assignment faces less lexical ambiguity than category + selector assignment while at the same time being harder to disambiguate via contexts.

Identifying the empty heads in the corpus has a profound effect on the inferrability of category features, nearly halving the number of ambiguous items. Even without doing this, however, category features are strongly ISL-2 inferrable for over 94% of all items, and over 75% of lexically ambiguous ones. When this is relaxed from strong to weak inferrability these numbers increase to 98% and 95% respectively. If empty heads are also identified, then category features become weakly inferrable for over 99% of all lexical items, and over 97% of lexically ambiguous ones. Our results therefore

				ISL-2 Ambig. Items			
Feature Set	Empty Heads Filled?	Total Items	ISL-1 Ambig. Subtotal	Strongly Ambig.		Weakly Ambig.	
Category Only	No	29610	8369	1762	(21.1%)	422	(5.0%)
Category Only	Yes	29685	8414	1210	(14.4%)	264	(3.1%)
Category + Selector	No	36635	18124	2861	(15.8%)	808	(4.5%)
Category + Selector	Yes	36688	18157	1571	(8.7%)	330	(1.8%)

Table 1: Count and percentage of lexical items which are ambiguous under each condition tested. Percentages of ISL-2 ambiguous items are calculated w.r.t. the number of ISL-1 ambiguous items as explained in the text.

Issue	Count	%
Wrong category	199	75.4%
Wrong category in other contexts	29	11.0%
Inconsistent category	10	3.8%
Non-alpha symbol	11	4.2%
Ambig. functional head complement	7	2.7%
Problem unclear	7	2.7%
Empty selector and complement	1	0.4%
Total	264	

Table 2: Reason for ambiguity of category features which are not weakly ISL-2 inferrable with identified empty heads.

show that both category features and selector features are largely ISL inferrable using contexts of size 2, which is in line with the conjecture that ISL inferrability is a restriction on category systems in human language.

4.2 Where ISL-2 inferrability fails

Of the subset of lexical items for which category features are not weakly ISL-2 inferrable (with identified empty heads), over 90% correspond to some error in the MGBank corpus. These included an incorrect category label on the lexical item in question, an incorrect category label on another lexical item with the same exponent, or general inconsistency in the category assigned to that form (i.e. one or the other should have been used uniformly). Table 2 summarizes the reasons why weak ISL-2 inferrability failed, with a count of the number of items affected when only category features are included and empty heads are identified. The first four reasons for ambiguity correspond to annotation problems in the corpus, while the rest reflect other reasons ISL-inferrability may have been difficult.

Looking more closely, the most common problem involved noun-noun compounds being misparsed as adjective-noun adjunction structures or vice versa. For example, “desktop computer” and “marketing director” were misparsed as adjective-noun sequences, while “imported steel” and “organized crime” were misparsed as noun-noun compounds. These errors alone accounted for nearly one third of the weakly ambiguous items. Similarly, the first word in a multi-word name like “Bloomfield Hills” or “West German” was occasionally misparsed as an adjective, adverb, or quantifier. In other cases, category for a given item was varied randomly between two reasonable choices. For instance, prenominal quantifiers were sometimes coded as ‘A’ and sometimes as ‘Q’. If the annotation had been consistent, the category would presumably have been recoverable. Overall, there were only a handful of items whose non-recoverability was not obviously related to annotation errors or empty heads.

Taken together, these results are promising for the ISL-inferrability conjecture: the category system used in MGBank displays a high degree of ISL-inferrability, and in cases where inferrability fails this is usually due to errors in the corpus itself.

5 Confounds and caveats

Our findings show that ISL-2 inferrability is an observable trend in MGBank. This could be taken as strong support of the conjecture of Graf (2020) that the category systems of natural languages are ISL-2 inferrable. However, there are several reasons why this might be too strong an inference.

5.1 The problem with adjunction

While ISL-2 inferrability looks like a plausible universal when considering heads and their arguments, it is much more likely to fail for adjuncts.

Consider a language like German, which makes a difference between adjectives and adverbs in

Feature Set	Empty Heads Filled?	Total Items	ISL-1 Ambig. Subtotal	ISL-2 Ambig. Items			
				Strongly Ambig.		Weakly Ambig.	
Movement Only	No	29456	7688	1407	(18.3%)	285	(3.7%)
Movement Only	Yes	29497	7708	469	(6.1%)	35	(0.5%)

Table 3: Count and percentage of lexical items which are ambiguous for movement features. Percentages of ISL-2 ambiguous items are calculated w.r.t. the number of ISL-1 ambiguous items as explained in the text.

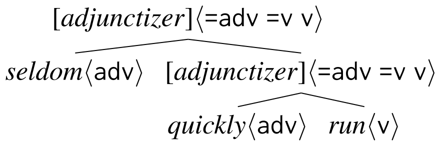


Figure 9: Adjunction as category-preserving selection. An adjunctizer head selects the adjunction site as its complement, and the adjunct itself as its specifier.

terms of distribution (and hence in terms of category features) but does not consistently mark this distinction in its morphology. Hence a form like *schnell* 'quick(ly)' could be an adverb or an adjective in a predicative construction like *Er ist schnell* 'he is fast'. In the analysis assumed by Graf (2020) and also here, adjuncts are modeled as arguments of an empty head – an *adjunctizer*. For example, an adverb adjoining to a VP would be modeled as the specifier of an empty V-head that takes a VP as its complement (Fig. 9). In such a configuration, the category of German *schnell* might not be ISL-2 inferrable. Its lower context would be $\ltimes$, and its upper context would be just the empty adjunction head and its complement, which might be yet another empty adjunction head. This context is equally compatible with *schnell* being an adjective or an adverb.

Since adjuncts are very common, even in corpora, it is suprising that we found such robust support for ISL-2 inferrability. Admittedly, over 40% of (weak and strong) ISL-2 inferrability failures for category feature in MGBank are on lexical items that are used (in at least some instance) as adjuncts, but many of those are related to coding errors. Given that there are theoretical reasons to doubt the viability of ISL-2 inferrability for a very common construction, there is reason to wonder whether the high rate of ISL-2 inferrability found in our study could be due to other confounds in the data.

5.2 Movement features as a control group

In contrast to category features, movement features represent syntactic relationships which are fundamentally non-local. There is no local way of predicting whether, say, an object is topicalized, on the basis of its arguments and seleting heads. Some features are more predictable, e.g. *which* is more likely to undergo wh-movement than remain in situ, and the C-head *do* is very likely to furnish a wh-landing site because of how *do*-support works in English. Still, theoretical considerations lead us to expect low ISL-2 inferrability scores for movement features. But, as shown in Table 3, the scores for movement features are very close to and sometimes even *better* than our findings in Section 4.1 for category (and category + selector) features.

We note that the distribution of movement features in the corpus is highly skewed, with over half of the movement-bearing lexical items being V heads with the +CASE feature. These are almost always dominated by empty transitive little-*v*, and select a DP argument — making +CASE highly inferrable. Even so, the finding is surprising from a theoretical perspective that focuses on what configurations are possible rather than which are common. In order to more accurately tease apart the factors contributing to inferrability, we turn to data simulations to provide a baseline.

5.3 Simulated data

Understanding whether ISL-inferrability is an intrinsic guiding principle of human language or simply a coincidence resulting from other properties requires setting up an appropriate baseline to test how much inferrability we might expect *without* this being an independent requirement of the system. To create such a baseline, synthetic datasets of lexical items and corresponding contexts were created programatically. These synthetic datasets are generated automatically based on i) the desired number of distinct exponents, ii) the desired num-

					ISL-2 Ambig. Items			
Feature Set	Total Items	Phono. Forms	Ctxs. Per item	ISL-1 Ambig. Subtotal	Strongly Ambig.		Weakly Ambig.	
Simulation (Category)	29685	24769	11.9	6007	108	(1.8%)	53	(0.9%)
Simulation (Category & Selector)	36688	24769	9.6	13961	245	(1.8%)	137	(1.0%)
Simulation (Movement)	29497	24769	12.0	5806	100	(1.7%)	50	(0.9%)

Table 4: Count and percentage of lexical items which are ambiguous in simulated data. Metrics of total lexical items, phonological exponents, and contexts per item follow those for each category set tested (with filled empty heads). Percentages of ISL-2 ambiguous items are calculated w.r.t. the number of ISL-1 ambiguous items as explained in the text.

ber of distinct lexical items, and III) the average number of contexts in which each lexical item appears. Given these, the synthetic data is generated using the following assumptions:

1. Each exponent appears in at least one lexical item.
2. Each lexical item appears in at least one context.
3. The frequency distribution of phonological items is Zipfian, both in terms of how many lexical items each exponent appears in and in terms of how frequently they are part of contexts for other items. In other words, a few exponents appear in many lexical items while most appear in very few.
4. The frequency distribution of lexical items is Zipfian. In other words, a few lexical items appear in many contexts, while most appear in very few.

For each of the feature sets for which we examined inferrability in MGBank, corresponding synthetic datasets were created with identical values for the number of exponents, lexical items, and average contexts per lexical item. We then tested ISL-inferrability in these synthetic datasets, running three simulations for each experiment and averaging results across the simulations.

The simulated datasets show a high degree of inferrability, comparable to what we find in the actual corpus. Table 4 shows the inferrability results for simulated datasets with metrics matched to the corpus data for each feature set tested (category, category + selector, and movement). These high inferrability rates demonstrate that the simple assumption of Zipfian distributions yields datasets where inferrability arises as an emergent property, rather than being a hard constraint on feature systems.

6 Conclusion

This paper uses the MG treebank (Torr, 2017) to evaluate the conjecture of Graf (2017) that syntactic categories are ISL-2 inferrable over the kind of dependency trees used with Minimalist Grammars. Intuitively, this conjecture states that the syntactic category of a lexical item can be inferred from its own surface form and/or the surface forms of its arguments and/or the surface form of the head it is an argument of. So though the conjecture is stated in very technical terms specific to MGs and subregular syntax, its relevance — and thus the import of our findings — extends to all syntactic formalisms that assume syntactic categories and selectional restrictions. Our analysis of MGBank largely supports the conjecture in Graf (2017) that category systems are ISL-2 recoverable: ISL-2 recoverability fails only for a small number of lexical items, and many of these cases are arguably due to coding errors in the corpus.

However, we also found a high degree of ISL-2 recoverability for movement features and category features of adjuncts, which is unexpected as neither kind of feature should be reliably ISL-2 inferrable. Through simulation, we also showed that a high level of inferrability can result simply from the frequency distribution of language datasets – namely, a Zipfian distribution.

Together, these findings indicate that human language category systems (and other syntactic features) are reliably ISL inferrable, but that this may not be due to a specific direct requirement for inferrability. In terms of Chomsky (2005), ISL-2 inferrability may be a third factor principle rather than a hard constraint of UG.

Regardless of the reason for which ISL-inferrability appears, its prevalence is a useful property of language to understand. One key benefit to identifying such properties is that they can often be leveraged for learning — just as many proposed language learning strategies leverage the Zipfian distributions that are known to be present. ISL-2 inferrability is particularly suggestive of an approach children may take in learning syntax. It offers a clear direction in which to *generalize*: two phonologically identical items in the same local context *must* also have the same category.

This work furnishes a proof-of-concept for the ISL-2 inferrability of syntactic features and suggests a method for further corpus work which might extend these results to more languages and data sources.

Acknowledgments

The work reported in this paper was supported by the National Science Foundation under Grant No. BCS-1845344 and the Institute for Advanced Computational Science at Stony Brook University. We thank the anonymous reviewers for their extensive comments and suggestions.

References

Noam Chomsky. 2005. Three factors in language design. *Linguistic Inquiry*, 36(1):1–22.

John Doner. 1970. Tree acceptors and some of their applications. *Journal of Computer and System Sciences*, 4:406–451.

Thomas Graf. 2011. Closure properties of Minimalist derivation tree languages. In *LACL 2011*, volume 6736 of *Lecture Notes in Artificial Intelligence*, pages 96–111, Heidelberg. Springer.

Thomas Graf. 2012. Locality and the complexity of Minimalist derivation tree languages. In *Formal Grammar 2010/2011*, volume 7395 of *Lecture Notes in Computer Science*, pages 208–227, Heidelberg. Springer.

Thomas Graf. 2013. *Local and Transderivational Constraints in Syntax and Semantics*. Ph.D. thesis, UCLA.

Thomas Graf. 2017. A computational guide to the dichotomy of features and constraints. *Glossa: a journal of general linguistics*, 2(1).

Thomas Graf. 2020. Curbing feature coding: Strictly local feature assignment. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 224–233.

Thomas Graf. 2022a. Diving deeper into subregular syntax. *Theoretical Linguistics*, 48:245–278.

Thomas Graf. 2022b. Subregular linguistics: bridging theoretical linguistics and formal grammar. *Theoretical Linguistics*, 48(3-4):145–184.

Gregory M. Kobele. 2011. Minimalist tree languages are closed under intersection with recognizable tree languages. In *LACL 2011*, volume 6736 of *Lecture Notes in Artificial Intelligence*, pages 129–144.

Gregory M. Kobele, Christian Retoré, and Sylvain Salvati. 2007. An automata-theoretic approach to Minimalism. In *Model Theoretic Syntax at 10*, pages 71–80.

Mitch Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Frank Morawietz. 2003. *Two-Step Approaches to Natural Language Formalisms*. Walter de Gruyter, Berlin.

James Rogers. 1998. *A Descriptive Approach to Language-Theoretic Complexity*. CSLI, Stanford.

James Rogers. 2003. wMSO theories as grammar formalisms. *Theoretical Computer Science*, 293:291–320.

Edward P. Stabler. 1997. Derivational Minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–95. Springer, Berlin.

Edward P Stabler. 2011. Computational perspectives on minimalism. *Oxford Handbook of Linguistic Minimalism*.

James W. Thatcher and J. B. Wright. 1968. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical Systems Theory*, 2(1):57–81.

Hans-Jörg Tiede and Stephan Kepser. 2009. Monadic second-order logic and transitive closure logics over trees. *Research on Language and Computation*, 7:41–54.

John Torr. 2017. Autobank: a semi-automatic annotation tool for developing deep minimalist grammar treebanks. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 81–86.

Appendix A: Compiled Results

Category Features

	Corpus		Simulation	
Type	Ambig. Items	% of Lexically Ambig. Items	Ambig. Items	% of Lexically Ambig. Items
SL1	8414	-	6007	-
SL2 (strong)	1210	14.4%	108	1.8%
SL2 (weak)	264	3.1%	53	0.9%

Total Items: 29,685 Phono. Forms: 24,769 Contexts per Item: 11.9

Table 5: Side-by-side comparison of inferrability results for category features (with filled empty heads) from the corpus study and the simulated data. Simulated data was generated using the number of items, exponents, and categories per item which matched the corpus (indicated at bottom of table).

Category & Selector Features

	Corpus		Simulation	
Type	Ambig. Items	% of Lexically Ambig. Items	Ambig. Items	% of Lexically Ambig. Items
SL1	18,157	-	13,961	-
SL2 (strong)	1571	8.7%	245	1.8%
SL2 (weak)	330	1.8%	137	1.0%

Total Items: 36,688 Phono. Forms: 24,769 Contexts per Item: 9.6

Table 6: Side-by-side comparison of inferrability results for category and selector features (with filled empty heads) from the corpus study and the simulated data. Simulated data was generated using the number of items, exponents, and categories per item which matched the corpus (indicated at bottom of table).

Movement Features

	Corpus		Simulation	
Type	Ambig. Items	% of Lexically Ambig. Items	Ambig. Items	% of Lexically Ambig. Items
SL1	7708	-	5806	-
SL2 (strong)	469	6.1%	100	1.7%
SL2 (weak)	35	0.5%	50	0.9%

Total Items: 29,497 Phono. Forms: 24,769 Contexts per Item: 12

Table 7: Side-by-side comparison of inferrability results for movement features only (with filled empty heads) from the corpus study and the simulated data. Simulated data was generated using the number of items, exponents, and categories per item which matched the corpus (indicated at bottom of table).

All Features

	Corpus		Simulation	
Type	Ambig. Items	% of Lexically Ambig. Items	Ambig. Items	% of Lexically Ambig. Items
SL1	19,493	-	15,314	-
SL2 (strong)	1832	9.4%	237	1.5%
SL2 (weak)	394	2.0%	133	0.9%

Total Items: 37,873 Phono. Forms: 24,769 Contexts per Item: 9.3

Table 8: Side-by-side comparison of inferrability results for entire feature string (with filled empty heads) from the corpus study and the simulated data. Simulated data was generated using the number of items, exponents, and categories per item which matched the corpus (indicated at bottom of table).

Appendix B: MGBank Categories

Category	**Num. Lexical Items**
n	19,585
v	9,574
adj	5,122
adv	964
lv	693
q	589
p	349
D	293
c	147
t	90
part	89
prog	64
mod	64
d	63
perf	32
voice	28
intj	23
tbar	22
negs	21
punc	13
prd	12
neg	12
log	8
ln	8
adjc	2
advc	2
vbar	2
self	1
features	1
top	1
Total	37,874

Table 9: Category features present in MGBank and the number of lexical items of each category.

Measuring the Impact of Segmental Deviation on Perceptions of Accentedness using Gradient Phonological Class Features

Nitin Venkateswaran
Department of Linguistics
University of Florida
venkateswaran.n@ufl.edu

Rachel Meyer
Department of Linguistics
University of Florida
rmeyer2@ufl.edu

Ratree Wayland
Department of Linguistics
University of Florida
ratree@ufl.edu

Abstract

Using Phonet (Vásquez-Correa et al., 2019), a neural network-based model, we generate vector representations of speech segments consisting of phonological class probabilities and use these representations to quantify segmental deviations in the English of native Hindi speakers from American English (AE) and Indian English (IE) baselines, in order to explain how these deviations impact perceptions of accentedness by native AE speakers. The primary focus is on three AE phonemes and their realizations in Hindi English (HE) and Indian English: the labiovelar approximant /w/, often produced as the labiodental approximant [ʋ]; the alveolar stop /t/, commonly realized as the retroflex stop [ʈ]; and the rhotic approximant /ɹ/, rendered as the rhotic tap [ɾ]. Multinomial logistic regressions of Euclidean distances from HE segments to AE/IE baselines on accent ratings show that larger distances from AE baselines increase the likelihood of perceiving stronger accents while larger distances from IE baselines decrease the likelihood. Changes in the probability distributions of contrastive phonological classes are found to correlate with the strength of the perceived accent. These results offer valuable insights into the interplay between native phonology and the perception of accented speech.

1 Introduction

The growing prevalence of English as a global *lingua franca* has led to a diverse variety of Englishes shaped by local linguistic and cultural influences. Among these, Indian English occupies a unique position, with distinct phonological characteristics arising from substrate Indo-Aryan and Dravidian languages (for more, see Wiltshire, 2020). These characteristics often include systematic phonetic differences, which are perceived as accented speech by speakers of other varieties of English.

This study explores how phonetic variation in Hindi English, i.e. the English of native Hindi speakers, influences perceptions of accentedness by native speakers of American English. We focus on three American English phonemes: the labiovelar approximant /w/, often produced as the labiodental approximant [ʋ] in Hindi English (Sailaja, 2009; Wiltshire and Harnsberger, 2006; CIEFL, 1972); the alveolar stop /t/, commonly realized as the retroflex stop [ʈ] (Masica, 1991; Kachru, 1986); and the rhotic approximant /ɹ/, rendered as the rhotic tap [ɾ] (Wiltshire, 2015; Krishnamurti, 2003; Masica, 1991). We use Phonet (Vásquez-Correa et al., 2019), a neural network based on Gated Recurrent Units (GRU) (Chung et al., 2014), to train a single model on large speech corpora of both American and Indian English to infer the classification probabilities of phonological classes associated with the phone segments of both Englishes. The resulting probability vectors are treated as representations of the phone segments in a joint vector space spanning both Englishes. These representations are used to examine the relationship between perceived accent and the Hindi English segments' proximity to American and Indian English baselines in the joint vector space. The segments [ʋ], [ʈ], and [ɾ] are produced uniformly in similar contexts across the varieties of Indian English spoken in the Indian subcontinent (Wiltshire, 2020), including the English of native speakers of Hindi and other Indo-Aryan languages (Fuchs, 2019; Sirsa and Redford, 2013; Wiltshire and Harnsberger, 2006); this facilitates the use of Indian English baselines to study variations in accent perception driven by these segments in Hindi English speaker productions. Quantifying the degree of accentedness using explainable probability vector representations could also facilitate an empirical validation of theories of second language speech learning, in particular the Speech Learning Model (SLM/SLM-r; Flege and Bohn 2021) and the Perceptual Assimilation Model (PAM; Best 1995); the joint vector space of the trained Phonet model could be surmised as a *perceptual space* of

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 222-234.
Eugene, Oregon, July 18-20, 2025

segment representations to test theories of speech learning, with distances/similarities between the representations serving as indicators of how second language learners might assimilate the phonetic categories of the language being learned into their own native categories.

2 Related Work

There are a number of studies that investigate accent classification and native language identification using corpora of spoken English from the Indian sub-continent, employing both handcrafted feature-based and neural network-based methods. These studies have used a variety of inputs such as MFCC-based features, prosodic features, formant frequencies, and raw spectrogram-based features with a range of classification models (Guntur et al., 2019; Krishna and Krishnan, 2014; Cheng et al., 2013; Sharma et al., 2024; China Bhanja et al., 2022; Siddhant et al., 2017; Jiao et al., 2016). Feature-based approaches offer explainable results at the expense of hand-crafting time- and resource-intensive features, and neural network approaches are black-box mechanisms capable of automatically deducing key features from the data input. We use Phonet to automatically convert key aspects of the spectral speech input into explainable vector representations of speech segments, thereby facilitating an explainable framework relating accent perception to gradient phonetic variation.

Other computational methods have been instrumental in capturing gradient phonetic variation which, unlike Phonet, have relied on traditional machine-learning approaches. For example, Yuan and Liberman (2009) introduced a method for capturing nuanced variations, such as degrees of /l/-darkness in American English, using log probability scores from forced alignments instead of categorical phone labels. This method, extended in later work (Yuan and Liberman, 2011), demonstrated both categorical distinctions and gradient degrees of /l/-darkness across contexts. Support Vector Machines have been used to classify r-full and r-less tokens in English using MFCCs (McLarty et al., 2019). Random forest classification has also been employed to model sociophonetic variables (Villarreal et al., 2020), estimating variable realizations by comparing acoustic features with canonical pronunciations.

Approaches that model phonological class probabilities—as done in Phonet—broaden the scope of analysis from individual segments to sets of segments that share articulatory or acoustic features. This shift enables a more generalized and interpretable analysis of speech, since phonological classes such as [continuant] and [sonorant] encode linguistically meaningful distinctions that underlie multiple segments. By modeling speech at the level of these classes, we capture gradient variation along perceptually and articulatorily relevant dimensions, facilitating cross-speaker and cross-context generalization. Moreover, class-based representations align with theoretical models of speech perception and learning, which emphasize feature-based similarity rather than segmental identity. As shown in Tang et al. 2023, such representations complement traditional acoustic measures and have proven effective in capturing phonetic processes like lenition (Wayland et al., 2023).

3 Methods

This section provides an overview of the Phonet model, its architecture and training methodology, the datasets used for its training, and the dataset consisting of the English of native Hindi speakers with accent annotations.

3.1 Phonet model

Phonet is a GRU-based neural network that estimates the posterior probabilities of the occurrence of phonological classes from speech signals. The signal is chunked into half-second segments, following which the log energy signal across 33 triangular filters along the Mel scale is calculated for each 25-ms window in the chunk. These log-energy feature sequences are processed by two bi-directional GRUs and a time-distributed dense layer, followed by separate dense layers for classifying each phonological class in a multi-task learning setup to calculate the probabilities of the classes associated with the input feature sequence. The probabilities are averaged across the frames to give a unique vector of the probabilities of phonological classes for each phone segment. The bi-directional GRU captures co-articulation effects by incorporating information from surrounding segments.

3.2 Phonological classes

Phonemes are grouped into phonological classes based on their shared phonetic features. One common distinction is between [+consonantal] and [-consonantal] phonemes. Consonantal phonemes, such as stops, fricatives, affricates, nasals, and liquids, involve constriction of the articulators in the vocal tract and are labeled [+consonantal]. In contrast, vowel and glide phonemes are typically labeled [-consonantal] because they do not involve the same level of constriction. An in-depth guide to phonological classes can be found in Hayes (2011). For the American and Hindi English phonemes in this study, the labiovelar approximant /w/ is defined by the classes [+sonorant, +continuant, +approximant, +voice, +round, +labial, +dorsal +high, +back, +tense], while the labiodental approximant /ʋ/ is defined by [+sonorant, +continuant, +approximant, +voice, -round, +labial, +labiodental, -dorsal, -high, -back]. The alveolar /t/ is [+consonantal, +coronal, +anterior], but the retroflex /ʈ/ is [+consonantal, +coronal, -anterior]. Finally, the approximant /ɹ/ is [-consonantal, +sonorant, +continuant, +approximant, -tap, +voice, +coronal, +distributed], while the tap /ɾ/ is [+consonantal, +sonorant, +continunant, +approximant, +tap, +voice, +coronal, -distributed, +anterior]. The classes that contrast the /w/-/ʋ/, /t/-/ʈ/, and /ɹ/-/ɾ/ pairs are of particular interest for analyzing against accent ratings.

3.3 Training datasets

To train models on American English and Indian English speech data, we use the English language datasets of the Mozilla Common Voice Speech Corpus (Ardila et al., 2020) and select datasets tagged with United States English and India and South Asia accent tags. Data from the Librispeech-100 corpus (Panayotov et al., 2015), the L2-ARCTIC non-native English speech corpus (Zhao et al., 2018), and the Indic Text-To-Speech (TTS) corpus (Baby et al., 2016) are used to source additional data in both Englishes. Only the English data from native Hindi speakers is selected from the L2-ARCTIC and Indic TTS datasets; however, the Mozilla Common Voice corpus does not include the speaker's native language tag for Englishes from the Indian sub-continent and all the data with the India and South Asia accent tag from this corpus is consequently used, forming the bulk of the training set for the Indian English data. A total of approximately 150 hours of American English and 120 hours of Indian English data are used for training.

3.4 Hindi English dataset with accent ratings

The CSLU FAE (Foreign Accented English) Release 1.2 dataset (Lander, 2007) contains continuous speech in English by speakers of 22 languages, including samples from native Hindi speakers. The corpus consists of telephone-quality utterances with information about perceptual judgments of the accents in the utterances. The speakers were asked to speak about themselves in English for 20 seconds. Three native speakers of American English independently listened to each utterance and judged the speakers' accents on a 4-point scale: *1-negligible/no accent*, *2-mild accent*, *3-strong accent* and *4-very strong accent*. To facilitate investigation of the drivers of accent perception relative to the *no/negligible* accent baseline, the minimum accent rating of the three speakers is taken as the aggregate rating for each recording. The *very strong* accent rating is subsequently merged into the *strong* one, given only one recording is tagged with that rating after applying the aggregate measure. Table 1 shows the distributions of the three accents across the recordings of native Hindi speakers, and Table 2 shows the distribution of the target Hindi English phone segments by accent rating and word position. We refer to this subset of the CSLU FAE dataset containing native Hindi speakers as the Hindi English dataset in subsequent sections.

3.5 MFA pre-processing

The Montreal Forced Aligner (MFA) tool (McAuliffe et al., 2017) is used to force-align the audio and transcripts of the training and Hindi English datasets, with the resulting TextGrid files used to label the phonological classes of each audio frame during Phonet training, in conjunction with the mapping of phone segments to phonological classes described in section 3.6. The transcripts are transcribed into IPA segments using the pre-trained MFA grapheme-to-phoneme (G2P) models and existing pronunciation dictionaries for American and Indian English (McAuliffe and Sonderegger, 2023a,b, 2024a,c). Custom acoustic models for American and Indian English are trained to avoid potentially noisy output from the existing pre-trained model (McAuliffe and Sonderegger, 2024b), given that this model is trained on a variety of world Englishes.

3.6 Phonet training and inference

To learn the phonological classes associated with phone segments during training, and to generate probability distributions over the classes for segments during inference, a mapping between the IPA segments in the MFA pronunciation dictionaries

Accent Rating	No. Recordings
No/Negligible	17
Mild	194
Strong	137
Total	348

Table 1: Distribution of accent ratings in the Hindi English dataset using a minimum aggregate of the ratings of three independent raters.

	Initial	Medial	Final
No/Negligible	29	31	44
Mild	294	346	376
Strong	246	264	256

(a) Distribution of [ʋ]

	Initial	Medial	Final
No/Negligible	23	50	86
Mild	138	569	957
Strong	120	374	643

(b) Distribution of [ʈ]

	Initial	Medial	Final
No/Negligible	12	15	14
Mild	115	173	179
Strong	76	121	157

(c) Distribution of [ɾ]

Table 2: Distribution of target segments in the Hindi English dataset by word position and accent rating.

and phonological classes is created for both American and Indian English phone sets. This mapping is created at the phonetic level, given that the learning of speech sounds in a second language occurs at the level of position-sensitive allophones and not at the phonemic level (Flege, 1995; Kohler, 1981).

A single Phonet model is trained on the combined American and Indian English training datasets to estimate the classification probabilities of phonological classes for segments of both languages in a joint vector space. The model can be said to incorporate the acoustic properties of both languages in its parameter weights; this means that, given a phone segment in the Hindi English data, the model can estimate whether the phonological class probabilities of that segment tend towards American English or Indian English baselines, or contain elements of both Englishes.

To facilitate joint training, the phone set to phonological class mappings of the two Englishes are merged into a single mapping, shown in Table 6 in the Appendix. The training and Hindi English datasets are force-aligned using the custom acoustic models described in Section 3.5. An 80-20 train-test split is used for training; the range of accuracy and F1 scores across the phonological classes can be found in Table 5 in the Appendix. The model is trained for a maximum of 30 epochs with early stopping, using the Adam optimizer (Kingma and Ba, 2014) with a categorical cross-entropy loss function.

3.7 Statistical Analyses

In the vector space of phonological class probabilities defined by the Phonet model, Euclidean distances are calculated between instances of the target Hindi English phone segments and the centroids of all instances of the baseline segments in the American and Indian English training data. The baselines consist of 500 recordings randomly sampled from each of the American and Indian English training datasets. The distances are regressed on the accent ratings using a multinomial logistic regression, taking the *no/negligible* rating as the reference level. The general hypotheses are that, relative to a *no/negligible* accent rating, the odds of a *mild* or *strong* accent should increase with increasing distance from the American English baseline and decrease with increasing distance from the Indian English baseline. Interactions of distance with word position are also investigated, given that variations in the categorization of a speech segment can be driven by the position of the segment in the word sequence (Dmitrieva, 2019). Two-way ANOVA tests are conducted to analyze the effect of accent rating and word position on the class probabilities of the Hindi English target segments. Significant differences would be expected for phonological classes that are contrastive between the baseline American English and target Hindi English segments, and the direction of the difference should correlate with differences in accent strength, suggesting that the class probabilities have an impact on the strength of the accent perceived. We report results only for those phonological classes which show significant main effects of accent ratings, or interaction effects of accent ratings with word position on the probabilities.

4 Results

Throughout this section, the terms AE and IE are used to refer to the American English and Indian English baselines respectively, with HE used to refer to the Hindi English dataset with accent ratings.

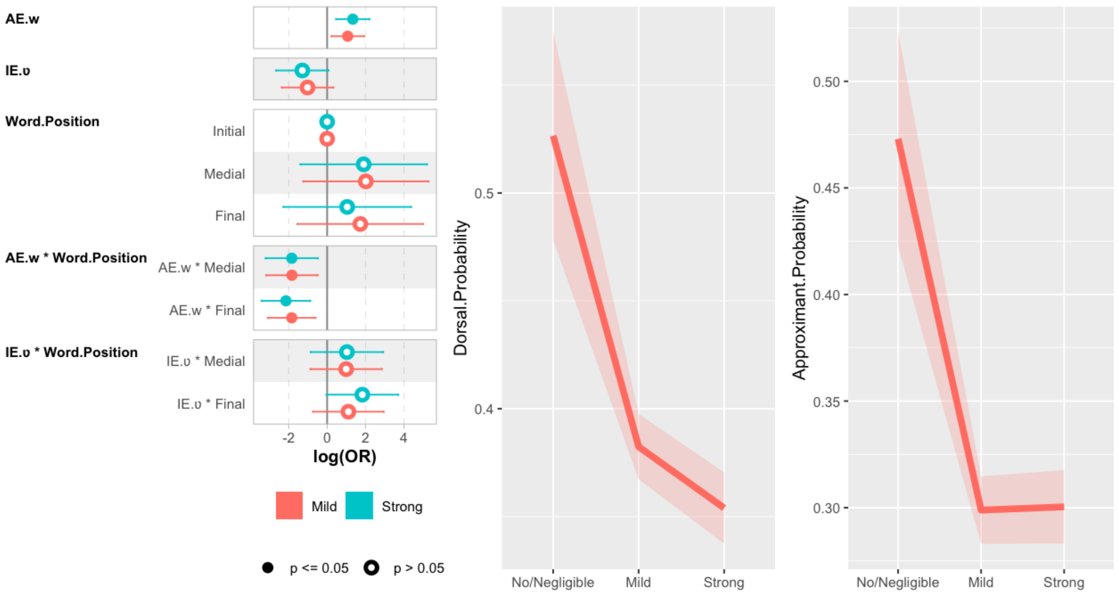


Figure 1: **Left:** Coefficient plots of multinomial logistic regression on accent ratings with reference level set to *no/negligible accent*, for the labiodental approximant [ʋ] in the Hindi English data. The interaction effect of Euclidean distance from AE [w] baseline with word position is significant, as is the main effect of distance from the AE baseline. **Center, Right:** Interaction plots of dorsal and approximant probabilities of the labiodental approximant [ʋ] in the Hindi English data by accent rating and initial word position (AE=American English; IE=Indian English).

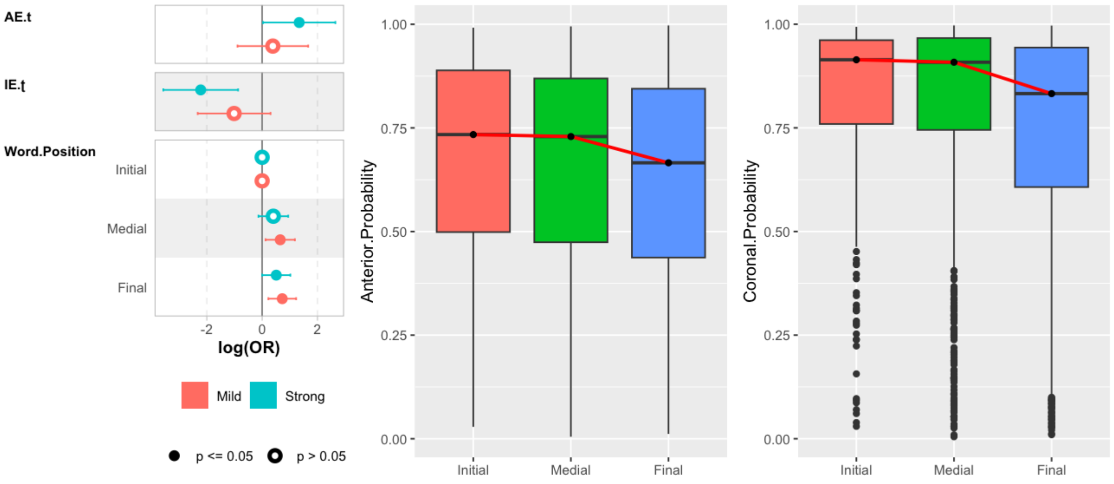


Figure 2: **Left:** Coefficient plots of multinomial logistic regression on accent ratings with reference level set to *no/negligible accent*, for the retroflex [ʈ] in the Hindi English data. The main effects of Euclidean distance from AE/IE baselines are significant, with increasing distance translating to higher/lower odds of strong accent perception. **Center, Right:** Distributions of anterior and coronal probabilities of retroflex [ʈ] in the Hindi English data by word position (AE=American English; IE=Indian English).

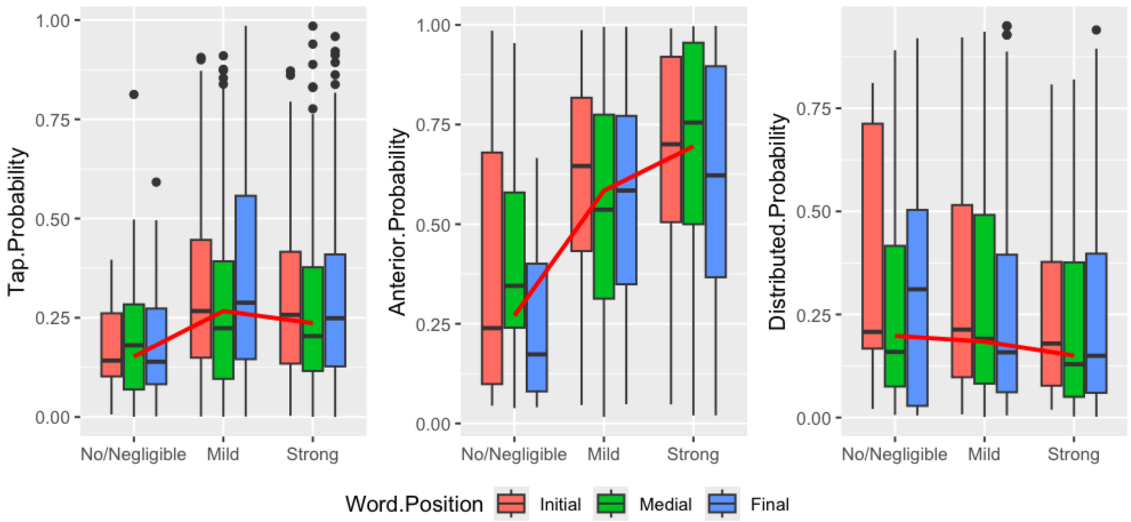


Figure 3: Distribution of tap, anterior, and distributed probabilities of rhotic tap [ɾ] in the Hindi English data by accent rating and word position. The differences in distributions across the accent ratings of all classes taken together suggest that speakers with the *strong* accent are producing the rhotic tap [ɾ] and those with *no/negligible* accent the rhotic approximant [ɹ].

Segment	Accent	Effect	β-coef.	*p*-val
[ʋ]	Mild	AE Dist.	1.069	.0144
		Medial Pos.	1.918	.233
		Final Pos.	1.479	.347
		AE*Medial	-1.833	.007
		AE*Final	-1.844	.0038
	Strong	AE Dist.	1.334	.0027
		Medial Pos.	1.79	.271
		Final Pos.	0.671	.674
		AE*Medial	-1.843	.008
		AE*Final	-2.146	.00093
[ʈ]	Mild	Medial Pos.	0.654	.0153
		Final Pos.	0.727	.0045
	Strong	AE Dist.	1.339	.0446
		IE Dist.	-2.22	.0013
		Final Pos.	0.510	.0497
[ɾ]	Mild	AE Dist.	3.567	9.2e-07
		IE Dist.	-3.041	1e-06
	Strong	AE Dist.	4.618	5.6e-09
		IE Dist.	-3.179	6.1e-07

Table 3: Log-odds coefficients (β-coef) of selected variables with accent rating as dependent, taking the *no/negligible* accent as reference level. Only significant effects are reported ($p< .05$). Positive log-odds coefficients suggest increased likelihood of the accent rating per unit increase in the regressor, relative to the reference accent. Negative coefficients suggest a decreased likelihood. (AE=American English; IE=Indian English).

4.1 Labiodental approximant [ʋ]

Figure 1 shows the coefficient plot of the multinomial logistic regression model described in Section 3.7, and Table 3 includes the β-coefficients for significant regressors with associated *p*-values. Interaction effects between distance from AE baseline and word position are significant both word medially and word finally. The main effect of distance from AE baseline is also significant. As Table 3 shows, for every unit increase in Euclidean distance from the AE baseline, the corresponding increase in the sum of the log-odds coefficients across main and interaction effects is higher word-initially and medially than word finally, suggesting higher odds of accent perception in these positions. There are no main nor interaction effects with distance from the IE [ʋ] baseline, suggesting that accent perception is driven by listeners' unmet expectations of perceiving the labiovelar approximant [w].

Looking at the two-way ANOVA tests, the interaction effects of accent rating and word position on dorsal and approximant probabilities are significant (dorsal: $F_{4,1877}$=3.121, p=.0143; approximant:$F_{4,1877}$=3.899, p=.0037). Tukey posthoc tests reveal significant differences in average dorsal probabilities word-initially between the *no/negligible* and *strong* accent ratings (p=.02), as well as significant differences in average approximant probabilities word-initially between the *no/negligible* and *mild* and *strong* accent ratings (*mild*: p=.0263; *strong*: p=.0315). The interaction plots in Figure 1 show that the dorsal and approximant probabilities decrease with increasing accent strength in word initial position, suggesting that speakers with stronger accents are using the labiodental instead of the labiovelar approximant.

4.2 Retroflex stop [ʈ]

Starting with the logistic regression, the results indicate that there are no significant interaction effects between distances from baselines and word position on accent ratings for the retroflex stop [ʈ]. There are significant main effects of distance from baselines for the *strong* accent rating (Table 3), with larger distance from AE/IE baseline resulting in higher/lower odds of the *strong* accent. Word position of the retroflex [ʈ] is significant medially and finally with the odds of perceiving an accent higher in those positions.

The two-way ANOVA tests show significant main effects of word position on both anterior ($F_{2,2951}$=5.327, *p*=.00491) and coronal ($F_{2,2951}$=25.980, *p*=6.6e-12) probabilities. Tukey post-hoc tests show lower average anterior probabilities word finally than in both initial (*p*=.02) and medial (*p*=.0397) positions, with word final coronal probabilities also lower than in initial (*p*<.001) and medial (*p*<.001) positions, as the probability distributions in Figure 2 show. However, there are no significant interaction effects word-finally between accent ratings and word position on the probabilities of either phonological class, nor are there significant main effects of accent ratings on the probabilities, suggesting that the anterior and coronal probabilities have no association with the strength of the accent rating for the retroflex [ʈ].

4.3 Rhotic tap [ɾ]

Results for the rhotic tap [ɾ] indicate that there are no interaction effects in the logistic regression between distances from baselines and word position. Significant main effects are observed for distance from baselines (Table 3), with larger distance from AE/IE baselines resulting in higher/lower odds of accent perception. The two-way ANOVA tests show significant main effects of accent ratings on anterior ($F_{2,853}$=26.08, *p*=1.02e-11), distributed ($F_{2,853}$=4.056, *p*=.0176) and tap ($F_{2,853}$=5.798, *p*=.00316) probabilities, and significant main effects of word position on tap probabilities ($F_{2,853}$=4.369, *p*=.01295). Tukey post-hoc tests reveal significant differences in average anterior probabilities between all accent rating pairs, with the largest differences between the *strong* and *no/negligible* (*p*<.001) and *mild* and *no/negligible* (*p*<.001) ratings. Differences in average distributed probabilities between *strong* and *mild* accent ratings are also significant (*p*=.03). Differences in tap probabilities between *mild* and *no/negligible* ratings are significant (*p*=.005) as well as between final and medial positions (*p*=.0093). These distributions are shown in Figure 3. Given that the tap, anterior and distributed classes between the tap [ɾ] and approximant [ɹ] rhotics are contrastive, when taken together the higher anterior and tap probabilities and lower distributed probabilities for *strong* and *mild* accents relative to the *no/negligible* accent could indicate that speakers in the HE dataset vary between the tap [ɾ] and the approximant [ɹ] in their productions, with strongly accented speakers tending towards the rhotic tap.

5 Discussion

5.1 Alignment with theories of second language speech learning

The results empirically show that instances of the Hindi English segments that are farther from the American (Indian) English baselines are associated with higher (lower) odds of an accent. These results align with predictions from contemporary theoretical models of cross-language speech learning, such as the Perceptual Assimilation Model (PAM; Best, 1995) and its extension (PAM-L2; Best and Tyler, 2007), which state that a second language learner's ability to perceptually distinguish speech categories in the language being learned (L2) depends on the categories' perceived similarity to the closest categories in the speaker's native language (L1). The Speech Learning Model (SLM; Flege, 1995) posits that learners at the initial stages of language learning subconsciously map L2 categories to their most similar L1 categories, and new L2 categories are eventually created in the learners' mental representations independent of their L1 categories as learners are exposed to more input distributions in the L2.

The existence of the labiovelar approximant [ʋ], retroflex stop [ʈ], and rhotic tap [ɾ] in the English of L1 Hindi speakers could be the result of transfer effects from learners' L1 language (Sharma, 2017; Kachru, 1986) or learners' exposure to productions from other speakers of Hindi English or Indian English (Sirsa and Redford, 2013). The transfer hypothesis is supported by the existence of the phonemic categories /ʋ/, /ʈ/ and /ɾ/ in Hindi, which also lacks the /w/, /t/ and /ɹ/ phonemes from General American English (Ohala, 1999; Masica, 1991; Giegerich, 1992). The realizations of the /w/,

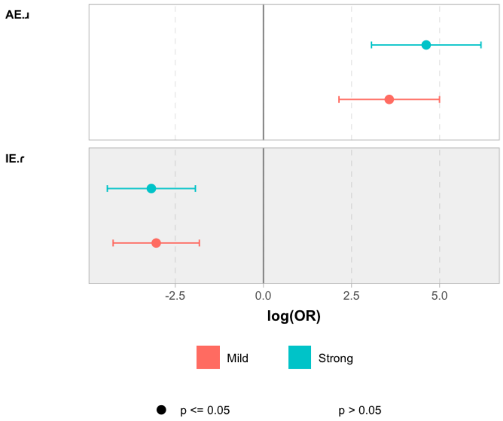


Figure 4: Coefficient plots of multinomial logistic regression on accent ratings with reference level set to *no/negligible* accent, for the rhotic tap [ɾ]. The main effects of Euclidean distance from AE/IE baselines are significant (AE=American English; IE=Indian English).

/t/ and /ɹ/ categories as [ʋ], [ʈ] and [ɾ] respectively in the Hindi English data are supported by the Single-Category assimilation model from PAM/PAM-L2, which predicts poor discrimination of the American English categories when they are perceived by learners to be similar to their L1 Hindi categories. The SLM also predicts the realization of the L1 Hindi categories in speech in place of the American English categories once learners subconsciously map the American English categories to their most similar L1 Hindi categories. To get an approximate similarity measure, the cosine similarities between the baseline American English categories and the L1 Hindi categories in the Hindi English data are computed in the joint vector space of the Phonet model, using their probability vector representations. Only the set of speakers with a *strong* accent rating is used for the calculation, given that speakers with *no/negligible* or *mild* accents may be producing American English-like categories in their speech in line with the SLM hypothesis described. The cosine similarities between the category pairs are strong ([w]-[ʋ]: μ=0.70, σ=0.14; [t]-[ʈ]: μ=0.81, σ=0.12; [ɹ]-[ɾ]: μ=0.74, σ=0.07), which supports the predictions of the PAM/PAM-L2 and SLM models.

Also consistent with the SLM model is the finding that the perceived degree of accentedness varies depending on the position of the segment within the word, as the mapping of L2 to L1 sounds occurs at the level of position-sensitive allophones. For example, larger distances from the American English labiovelar approximant [w] baseline are more prominent word-initially and medially, and the retroflex [ʈ] segment has a greater impact on accentedness perception word-medially and finally, possibly because the category /t/ is realized in American English as retroflex [ʈ] primarily in word-initial positions and particularly before the rhotic approximant [ɹ] as in ’try’ (Polka, 1991).

The retroflex [ʈ] segments in word-final position in the Hindi English data have lower anterior and coronal probabilities than in initial and medial positions, suggesting a higher degree of retroflexion word-finally. The lack of significant effects of accent ratings on anterior and coronal probabilities, together with the significant effect of word-final position on accent strength and the high degree of word-final retroflexion suggest that while the production of the retroflex [ʈ] segment is significant, there may be other acoustic differences between the [t]/[ʈ] segments that are more salient to the perception of accentedness. This finding lines up with research showing that American English speakers have difficulty distinguishing retroflex from dental stops in Hindi (Pruitt et al., 2006; Polka, 1991), suggesting a lack of sensitivity to retroflexion.

The significant difference in average dorsal and approximant probabilities between the *no/negligible* and *strong* accents for the labiodental approximant [ʋ] segments suggests that English speakers of Hindi realize the segment as a labial sound without the accompanying tongue back approximation toward the velum. Moreover, the constriction at the lips is too narrow to achieve the typical resonance of an approximant. For the rhotic tap [ɾ] segment, higher anterior and tap probabilities for *mild* and *strong* accents indicate a forward articulation consistent with a tap rather than the retracted, posterior articulation of the American English [ɹ]. Lower distributed probabilities for *mild* and *strong* accents suggest a reduced tongue contact spread, characteristic of the localized articulation of the tap and contrasting with the broader tongue configuration of the approximant [ɹ].

5.2 Investigating Phonet’s probability-based representations for accent classification

We investigate whether the phonological class probability vectors generated by Phonet for the segments in this study can differentiate among accent ratings relative to two baseline representations: the log Mel-filterbank (MFCC) transformations described in Section 3.1 that serve as input to the Phonet model, and pre-trained embeddings from the final transformer layer of the WavLM architecture, using the `wavlm-large` model (Chen et al., 2022). The MFCC and WavLM representations are derived by averaging across all frames for the segment. We run two types of accent classification models that take the representations as input: a linear support vector classifier (SVC) with L2 regularization, with a cross-validated grid search determining the optimal regularization parameter, and a neural network classifier (NNet) with a single dense layer of size 512 that uses a ReLU activation, followed by a softmax classification layer. All neural network models are trained using the categorical cross-entropy loss with the Adam optimizer default parameters and a dropout value of 0.5. The Phonet probabilities, like the MFCC representations, are log-transformed. An 80-20 train-test split is used with results averaged across three seeds.

		***F*-score**	
Segment	**Features**	**SVC**	**NNet**
[ʋ]	MFCC	51.28	51.74
	Phonet	45.93	52.43
	WavLM	62.14	68.34
[ʈ]	MFCC	50.19	47.64
	Phonet	49.96	52.44
	WavLM	69.96	79.3
[ɾ]	MFCC	52.56	57.41
	Phonet	52.39	55.65
	WavLM	61.37	67.24

Table 4: F-scores from linear support vector (SVC) and neural network (NNet) based accent classifiers using features from different segment representations as input. Results are averaged across three seeds.

The results in Table 4 show that the WavLM representations, as expected, discriminate the accent ratings best across all segments and classifier types. The nonlinear neural network classifiers trained using Phonet representations show noticeable improvements in the F-score across all segments when compared to the linear SVC classifiers. The improvement is particularly visible with the labiodental approximant [ʋ]: the biased linear SVC classifier does worse with Phonet representations compared to MFCC-based ones whereas the nonlinear neural network classifier shows comparable performance between the two representations. The MFCC-based neural network classifiers, in contrast, only show improvement over the linear SVC classifiers for the rhotic tap [ɾ] segment, with worse results for the retroflex [ʈ] segment possibly due to overfitting. These findings indicate that the Phonet-based representations may be richer than the MFCC-based ones in the sense that they may contain more nonlinear relationships and interactions that can be unlocked by more complex models; however, they do not rival the pre-trained WavLM representations which contain more information to better discriminate accents, at the cost of reduced explainability.

6 Conclusion and Future Directions

This study demonstrates the use of a neural network model, Phonet, to capture gradient phonetic variation revealing nuanced patterns of L2 mispronunciation that align with and extend second-language speech theories. These findings align with theoretical models of second language speech learning such as the Perceptual Assimilation Model and the Speech Learning Model, particularly in demonstrating the influence of L1 phonological systems on L2 production and the positional sensitivity of speech articulation. The study highlights how gradient phonetic variation offers deeper insights into the articulatory and perceptual mechanisms underlying accentedness, bridging theoretical predictions and empirical observations. Beyond validating second-language speech models, this approach unveils fine-grained articulatory details, advancing our understanding of L2 speech learning and providing a robust foundation for future research in cross-language speech perception and production. Future research could explore observed patterns of L2 English mispronunciation and positional sensitivity for other L1 languages using pre-trained model representations to see if similar generalizations emerge. Analyzing co-articulatory effects and dynamic speech variations could further bridge theoretical models and real-world speech patterns, offering deeper insights into second-language acquisition.

Acknowledgments

This research is partially funded by NSF-National Science Foundation (SenSE). Award No. 852 2037266 – SenSE. We also thank the reviewers for their helpful feedback.

References

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common Voice: A massively-multilingual speech corpus. In *Proceedings of the*

Twelfth Language Resources and Evaluation Conference, pages 4218–4222, Marseille, France. European Language Resources Association.

Arun Baby, Anju Leela Thomas, N. L. Nishanthi, and TTS Consortium. 2016. Resources for Indian languages. In *CBBLR – Community-Based Building of Language Resources*, pages 37–43, Brno, Czech Republic. Tribun EU.

C. T. Best and M. D. Tyler. 2007. Nonnative and second-language speech perception: Commonalities and complementarities. In M. J. Munro and O. S. Bohn, editors, *Language experience in second language speech learning: In honor of James Emil Flege*, pages 13–34. Amsterdam:Benjamin.

Catherine T. Best. 1995. *A Direct Realist View of Cross-Language Speech Perception*. Speech perception and linguistic experience: Issues in cross-language research, Strange, Winifred [Ed], Timonium, MD: York Press, Inc, 1995, pp 171-204. York Press, Inc.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. 2022. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Jian Cheng, Nikhil Bojja, and Xin Chen. 2013. Automatic accent quantification of Indian speakers of English. In *Interspeech*, pages 2574–2578.

Chuya China Bhanja, Mohammad Azharuddin Laskar, Rabul Hussain Laskar, and Sivaji Bandyopadhyay. 2022. Deep neural network based two-stage Indian language identification system using glottal closure instants as anchor points. *Journal of King Saud University - Computer and Information Sciences*, 34(4):1439–1454.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

CIEFL. 1972. *The Sound System of Indian English*. CIEFL, Monograph 7. Hyderabad.

Olga Dmitrieva. 2019. Transferring perceptual cue-weighting from second language into first language: Cues to voicing in Russian speakers of English. *Journal of Phonetics*, 73:128–143.

James E. Flege. 1995. Second language speech learning: Theory, findings, and problems. *Speech Perception and Linguistic Experience: Issues in Cross-Language Research*, 92:233–277.

James Emil Flege and Ocke-Schwen Bohn. 2021. *The Revised Speech Learning Model (SLM-r)*, page 3–83. Cambridge University Press.

Robert Fuchs. 2019. Almost [w] anishing: The elusive/v/-/w/contrast in educated indian english. In *Proceedings of the 19th International Congress of Phonetic Sciences, Melbourne, Australia*, pages 1382–1386.

Heinz J. Giegerich. 1992. *English Phonology: An Introduction*. Cambridge Textbooks in Linguistics. Cambridge University Press.

Radha Krishna Guntur, Krishnan Ramakrishnan, and Vinay Kumar Mittal. 2019. Non-native Accent Partitioning for Speakers of Indian Regional Languages. In *Proceedings of the 16th International Conference on Natural Language Processing*, pages 65–74, International Institute of Information Technology, Hyderabad, India. NLP Association of India.

Bruce Hayes. 2011. *Introductory Phonology*. John Wiley & Sons.

Yishan Jiao, Ming Tu, Visar Berisha, and Julie M Liss. 2016. Accent Identification by Combining Deep Neural Networks and Recurrent Neural Networks Trained on Long and Short Term Features. In *Interspeech*, pages 2388–2392.

Braj B Kachru. 1986. The Indianization of English. *English Today*, 2(2):31–33.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

K. Kohler. 1981. Contrastive phonology and the acquisition of phonetic skills. *Phonetica*, 38:213–226.

G Radha Krishna and Raghava Krishnan. 2014. Influence of mother tongue on English accent. In *Proceedings of the 11th International conference on Natural Language Processing*, pages 63–67.

Bhadriraju Krishnamurti. 2003. *The Dravidian Languages*. Cambridge University Press, Cambridge.

T. Lander. 2007. *CSLU: Foreign Accented English Release 1.2 LDC2007S08. Web Download*. Linguistic Data Consortium, Philadelphia.

Colin Masica. 1991. *The Indo-Aryan languages*. Cambridge University Press, Cambridge.

M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger. 2017. Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi. In *Proc. Interspeech 2017*, pages 498–502.

Michael McAuliffe and Morgan Sonderegger. 2023a. English (India) MFA G2P model v3.0.0. Technical report.

Michael McAuliffe and Morgan Sonderegger. 2023b. English (US) MFA G2P model v3.0.0. Technical report.

Michael McAuliffe and Morgan Sonderegger. 2024a. English (India) MFA dictionary v3.1.0. Technical report.

Michael McAuliffe and Morgan Sonderegger. 2024b. English MFA acoustic model v3.0.0. Technical report.

Michael McAuliffe and Morgan Sonderegger. 2024c. English (US) MFA dictionary v3.1.0. Technical report.

Jason McLarty, Taylor Jones, and Christopher Hall. 2019. Corpus-Based Sociophonetic Approaches to Postvocalic R-Lessness in African American Language. *American Speech*, 94(1):91–109.

Manjari Ohala. 1999. Hindi. In *Handbook of the International Phonetic Association*, pages 100–103. Cambridge University Press, Cambridge.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

Linda Polka. 1991. Cross-language speech perception in adults: Phonemic, phonetic, and acoustic contributions. *The Journal of the Acoustical Society of America*, 89(6):2961–2977.

John S. Pruitt, James J. Jenkins, and Winifred Strange. 2006. Training the perception of Hindi dental and retroflex stops by native speakers of American English and Japanese. *The Journal of the Acoustical Society of America*, 119(3):1684–1696.

Pingali Sailaja. 2009. *Indian English*. Edinburgh University Press, Edinburgh.

Ch. Rahul A. N. Sharma, Harsh Kumar Singh, H.Suhas Prabhu, Aniketh V. Jambha, C Jyotsna, and Peeta Basa Pati. 2024. Accent Detection in Indian Languages through Convolutional Neural Network based Spectrogram Analysis. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–5.

Devyani Sharma. 2017. *Chapter 16: English in India*, pages 311–329. De Gruyter Mouton, Berlin, Boston.

Aditya Siddhant, Preethi Jyothi, and Sriram Ganapathy. 2017. Leveraging native language speech for accent identification using deep Siamese networks. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 621–628.

Hema Sirsa and Melissa A. Redford. 2013. The effects of native language on Indian English sounds and timing patterns. *Journal of Phonetics*, 41 6:393–406.

Kevin Tang, Ratree Wayland, Fenqi Wang, Sophia Vellozzi, Rahul Sengupta, and Lori Altmann. 2023. From sonority hierarchy to posterior probability as a measure of lenition: The case of Spanish stops. *The Journal of the Acoustical Society of America*, 153(2):1191–1203.

D. Villarreal, L. Clark, J. Hay, and K. Watson. 2020. From categories to gradience: Auto-coding sociophonetic variation with random forests. *Laboratory Phonology*, 11(1):6.

J.C. Vásquez-Correa, Philipp Klumpp, Juan Rafael Orozco-Arroyave, and Elmar Nöth. 2019. Phonet: A Tool Based on Gated Recurrent Neural Networks to Extract Phonological Posteriors from Speech. In *Interspeech 2019*, pages 549–553.

Ratree Wayland, Kevin Tang, Fenqi Wang, Sophia Vellozzi, and Rahul Sengupta. 2023. Quantitative acoustic versus deep learning metrics of lenition. *Languages*, 8(2).

Caroline Wiltshire. 2015. Dravidian varieties of Indian English. In G. K. Panikkar, B. Ramakrishna Reddy, K. Rangan, and B. B. Rajapurohit, editors, *Studies on Indian Languages and Cultures (V. I. Subramoniam Commemoration Vol. II)*, pages 49–63. International School of Dravidian Linguistics: Thiruvananthapuram.

Caroline R. Wiltshire. 2020. *Uniformity and Variability in the Indian English Accent*. Elements in World Englishes. Cambridge University Press.

Caroline R. Wiltshire and James D. Harnsberger. 2006. The influence of Gujarati and Tamil L1s on Indian English: A preliminary study. *World Englishes*, 25(1):91–104.

J. Yuan and M. Liberman. 2009. Investigating /l/ variation in English through forced alignment. In *Proc. Interspeech 2009*, pages 2215–2218.

Jiahong Yuan and Mark Liberman. 2011. /l/ variation in American English: A corpus approach. *Journal of Speech Sciences*, 1(2):35–46.

G. Zhao, S. Sonsaat, A. Silpachai, I. Lucic, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna. 2018. L2-ARCTIC: A Non-native English Speech Corpus. In *Proc. Interspeech 2018*, pages 2783–2787.

A Appendix

A.1 Phonet accuracy and F1 scores

Table 5 shows the Phonet model's accuracy and F1 classification scores for each phonological class.

A.2 Phone to phonological class mapping

Table 6 shows the merged mapping between the MFA phonesets from McAuliffe and Sonderegger (2024a,c) and the phonological classes from Hayes (2011).

Phonological Class	**Accuracy**	**F1 score**
syllabic	91.07	91.23
consonantal	91.55	91.59
long	86.69	88.8
sonorant	93.68	93.68
continuant	92.50	92.50
delayed release	91.98	92.57
approximant	92.86	92.9
tap	97.31	98.33
nasal	91.83	92.98
voice	93.2	93.2
spread glottis	95.66	96.81
labial	87.65	88.8
round	90.4	92.42
dental	96.15	97.33
coronal	88.65	89.02
anterior	88.08	88.79
distributed	87.56	90.31
strident	95.11	95.52
lateral	92.9	94.8
dorsal	90.97	91.01
high	87.56	88.61
low	91.37	92.41
front	90.26	90.99
back	90.33	92.01
tense	86.84	90.98
constr glottis	99.99	99.99

Table 5: Accuracy and F1 scores for classification of phonological classes by the Phonet model.

Phonological Class	**Phone List**
syllabic	a aj aw aː eː ej i iː oː ow æ ɐ ɑ ɑː ɒ ɒː ɔj ə ɚ ɛ ɛː ɜ ɜː ɝ ɪ ʉ ʉː ʊ
consonantal	b bʲ c cʰ cʷ d ʤ dʲ d̪ f fʲ h j k kʰ kʷ l m mʲ m̩ n n̩ p pʰ pʲ pʷ s t ʧ tʰ tʲ tʷ t̪ v vʲ z ç ð ŋ ɖ ɟ ɟʷ ɡ ɡʷ ɫ l̩ ɱ ɲ ɾ ɾʲ r̃ ʃ ʈ ʈʲ ʈʷ ʎ ʒ ʋ ʔ θ
long	aː ɑː ɒː iː ɛː ɜː eː oː ʉː
sonorant	a aː aj aw ɐ æ ɑ ɑː ɒ ɒː ɛ ɛː ɜ ɜː ɝ eː ej ɪ i iː oː ow ɔj ʉ ʉː ʊ ə ɚ l ɫ l̩ ʎ j ɾ ɾʲ r̃ ɹ m m̩ mʲ ɱ ŋ ɲ n̩ n ʋ w
continuant	a aː aj aw ɐ æ ɑ ɑː ɒ ɒː ɛ ɛː ɜ ɜː ɝ eː ej ɪ i iː oː ow ɔj ʉ ʉː ʊ ə ɚ ð θ f fʲ j ɾ r̃ ɾʲ ɹ ʃ ʒ v vʲ ç l ɫ l̩ ʎ h s z ʋ w
delayed release	f fʲ ʃ ʒ ç v vʲ ʧ ʤ h s z ð θ
approximant	a aː aj aw ɐ æ ɑ ɑː ɒ ɒː ɛ ɛː ɜ ɜː ɝ eː ej ɪ i iː oː ow ɔj ʉ ʉː ʊ ə ɚ j r̃ ɾ ɾʲ ɹ l l̩ ɫ ʎ ʋ w
tap	ɾ r̃ ɾʲ
nasal	m mʲ m̩ ɱ n n̩ ŋ ɲ
voice	a aː aj aw ɐ æ ɑ ɑː ɒ ɒː ɛ ɛː ɜ ɜː ɝ eː ej ɪ i iː oː ow ɔj ʉ ʉː ʊ ə ɚ ð d dʲ ɖ d̪ ɾ r̃ ɾʲ ɹ j ɟ ɟʷ ʒ ʤ v vʲ m mʲ m̩ ɱ n ŋ n̩ ɲ b bʲ l l̩ ʎ ɡʷ ɡ z ʋ w
spread glottis	h
labial	p pʲ pʰ pʷ f fʲ v vʲ ʋ ɱ mʲ m m̩ b bʲ
round	ɒ ɒː ow oː ɔj ʉ ʉː ʊ
dental	t̪ d̪ ð θ
coronal	c cʰ cʷ ç ɾ r̃ ɾʲ ɹ ʃ ʒ ʤ ʧ ʈ ʈʲ ʈʷ t tʰ tʷ tʲ t̪ n̩ n ɲ ʎ d ɖ d̪ dʲ l ɫ l̩ s z θ ð
anterior	ɾ r̃ ɾʲ t tʷ tʰ tʲ t̪ d dʲ d̪ n̩ n l ɫ l̩ s z θ ð
distributed	c ç cʰ cʷɟ ɟʷ ʧ ʤ ʃ ʒ ɹ ʎ ɲ θ ð
strident	s z ʧ ʤ ʃ ʒ
lateral	l ɫ l̩ ʎ
dorsal	a aː aj aw ɐ æ ɑ ɑː ɒ ɒː ɛ ɛː ɜ ɜː ɝ eː ej ɪ i iː oː ow ɔj ʉ ʉː ʊ ə ɚ c cʰ cʷ ç k kʷ ɡ ɡʷ ŋ ɲ ɫ l̩ ʎ w
high	ɪ i iː ʉ ʉː ʊ c cʰ cʷ ç k kʷ ɡ ɡʷ ʎ ŋ ɲ w
low	a aː aj aw ɑ ɑː ɒ ɒː æ
front	æ ɛ ɛː ɪi iː c cʰ cʷ ç eː ej j ɟ ɟʷ ɲ ʎ
back	ɑ ɑː ɒ ɒː ɜ ɜː ɝ oː ow ɔj ʊ ɫ l̩ w
tense	eː ej i iː ʉ ʉː oː ow ə ɚ j w
constr. glottis	ʔ

Table 6: Mapping between MFA phonesets and Hayes' phonological classes for Phonet modeling.

Left-corner Minimalist parsing of mixed word order preferences

Lei Liu
Institute of Linguistics
Leipzig University
lei.liu@uni-leipzig.de

Abstract

This paper proposes a uniform, structure-based account for mixed word order preferences crosslinguistically. These preferences include the short-before-long preference in the English heavy NP shift, the long-before-short preference in the Japanese transitive sentences, and the absence of word order preference in Mandarin Chinese preverbal PPs. The syntactic structures of each competing word orders are formally characterized using Minimalist grammars (MGs) and constructed with a left-corner MG parser. Complexity metrics are derived from the parser's behavior, which relate the difficulties of the structure building process to memory load. The metrics show that the preferred word orders are less memory-intensive to build than their counterparts in both the short-before-long and the long-before-short cases, while no memory resource differences are found for the case where no word order preference exists. The results suggest that the preferred word orders – or a lack thereof – follow from their syntactic structures. This further supports the viability of left-corner MG parsing as a psycholinguistically adequate model for human sentence processing.

1 Introduction

Word preferences are conditioned by at least two factors: a general efficiency principle to minimize dependency length and language-specific syntactic characteristics. The efficiency principle reflects the tendency of grammars to minimize the dependency lengths between syntactic elements. This principle takes the form of Dependency Length Minimization (DLM, Hawkins 1994, 2004) when focusing on the lengths of syntactic dependency relations; and as the Dependency Locality Theory (DLT, Gibson 2000) when focusing on the memory resource required to hold those dependencies. Prior research has shown that this efficiency principle accounts for the short-before-long order in head-initial languages (e.g., Wasow, 2002) and the long-before-short preference in head-final languages (e.g., Hawkins, 1994)

The second factor conditioning word order preferences, language-specific syntactic characteristics, helps explain word preference variations across languages. For example, Liu (2020) notes that the association between headedness and word order preference does not always hold crosslinguistically. Other language-specific properties should therefore be considered in understanding word order preferences. These include the degree of word order flexibility, the prominence of NPs (Yamashita and Chang, 2001) and the richness of the case marking system (Futrell et al., 2020), all of which interact with broader structural tendencies to shape observed preferences.

Despite fruitful results and increasing empirical coverage of the research on the two factors, the interplay between the efficiency principle and language-specific syntactic characteristics remains puzzling. One key issue is that it is unclear what syntactic features and in what ways affect the preference for DLM. Research on DLM often relies on dependency grammar as the description of syntax and measures dependency length in terms of the number of intervening words. While this approach is simple and effective for large-scale corpus studies, it may overlook important syntactic information that contributes to word order preferences. For example, Liu (2008) argues that in a language such as Chinese, the richness of functional words might add extra distance to heads and their dependents when compared to a language such as English, where the grammatical functions are realized by inflection. This accounts for the larger mean dependency distance of Chinese. However, it remains unclear whether it is the additional morphemes themselves in Chinese, the different syntactic processes these functional heads undergo, or the syntactic structure they occupy, that contributes

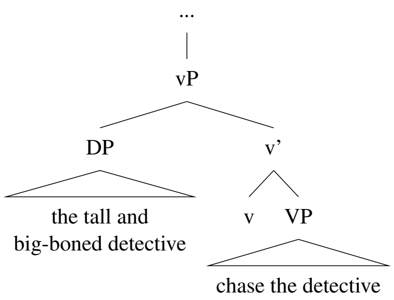
Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 235-243.
Eugene, Oregon, July 18-20, 2025

to the dependency length difference.

This paper aims to address the interplay of the general efficiency principle and specific syntactic characteristics in predicting word order preferences from a Minimalist parsing perspective. Minimalist parsing is particularly well-suited for this task because its complexity metrics rigorously relate detailed syntactic structures to a general processing constraint: memory resources. Kobele et al. (2013) measure memory resources associated with a top-down MG parser using *tenure*, the amount of time a tree node is retained in memory. The authors argue that *tenure* can be viewed as a generalization of the DLT principles which correlates processing difficulties with memory space needed for holding dependency relations. They show that *tenure*-based complexity metrics are shown to successfully model processing contrasts between verb clusters in Dutch and German, and center and right embeddings in English. Recent work has expanded the empirical coverage of this MG processing modeling program (e.g., stacked relative clauses in Mandarin and English Zhang 2017; attachment ambiguity in English and Korean Lee 2018; gradient difficulty in Italian relative clauses De Santo 2019, 2020; end-weight preference in English and Mandarin Liu 2022, among others).

One limitation of the top-down MG processing model is that it encounters difficulty capturing the long-before-short preference in Japanese transitive sentence (Liu, 2022, 2023). Intuitively, word order preferences arise when speakers try to order long constituents around other shorter ones to ease processing. This shows up in syntactic trees as unbalanced sister nodes. For instance, in an English sentence *The tall and big-boned detective chased the suspect*, the subject and the vP is a pair of unbalanced sister nodes, as shown in (1).

(1)



When no other syntactic operations are involved, the top-down parser explores the structure top-down and from left to right to follow the word order. After the parser expands vP to DP and v', exploring either branch requires the parser to store the other branch in memory. This makes exploring the less complex branch more memory-efficient, which is the intuition behind the short-before-long preference. And in order to derive the opposite order preference in Japanese, additional structural assumptions are needed, which presents a challenge to the model (Liu, 2022, 2023).

Against this background, we opt for the left-corner parser for MGs in this study. We argue that the left-corner Minimalist parsing model effectively captures the short-before-long, the long-before-short preferences, and the absence of order preference. According to the modeling results, the preferred word orders require fewer memory resources to build than their counterparts. Furthermore, no memory load difference is found for structures that do not exhibit order preferences.

The remainder of the paper proceeds as follows. Section 2 introduces Minimalist Grammars (MGs), a left-corner MG parser, and the key complexity metrics for our parsing model. Section 3 presents modeling results of the three word order preferences. Section 4 concludes the paper with a discussion on the role of syntactic assumptions in the parsing model.

2 Left-corner Minimalist parsing

The left-corner Minimalist parsing approach to processing modeling consists of three components: characterizing syntactic proposals using Minimalist Grammars (MGs), incorporating the formalisms into left-corner parsing models, evaluating modeling results based on complexity metrics connecting parsing difficulty to memory load.

Minimalist Grammar is chosen as the formalism for two reasons. First, it incorporates the toolbox needed for Chomskyan syntax, providing detailed structural information known to influence processing. Second, MG parsers are available and relatively well-understood from previous studies (top-down MG parsing: Stabler 2013; Kobele et al. 2013, left-corner MG parsing: Stanojević and Stabler 2018; Hunter et al. 2019).

A left-corner MG parser is used instead of a top-down parser to overcome known difficulties of the latter as discussed above. The left-corner MG parser, on the other hand, has been recently argued to be a plausible model for human sentence processing (Liu, 2024).

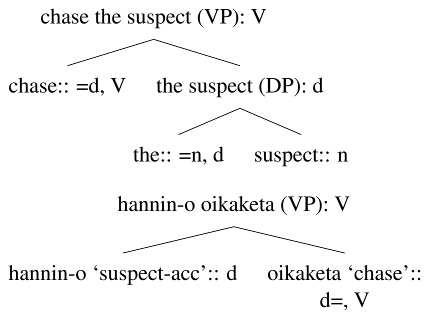
The following subsections introduce the grammar formalism and its left-corner parser, and the key complexity metric needed for the subsequent modeling work.

2.1 Minimalist Grammar and left-corner MG parser

Minimalist Grammar (MG, Stabler 1997, 2011) is a lexicalized, context-sensitive grammar formalism based on the Minimalist Program (Chomsky, 2014). In MGs, lexical items (LIs) are finite sequences of features containing information about sound, word shapes, and instructions for structure building operations. The grammar makes use of two such operations, `merge`, which combines categories, and `move`, which regulates movements.

`Merge` happens when two LIs have matching selector-selectee features as their first features. (2) illustrates how `Merge` builds a VP in English and Japanese.

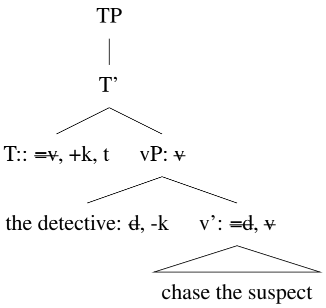
(2) a. b.



To build the VP, the objects bear the same selectee feature `d` in both the English and the Japanese cases. The selector feature of the verb is =`d` in English and `d`= in Japanese. The placement of the equal sign (=) indicates the selectee to be merged on the left or the right. This allows our model to capture headedness.

`Move` happens when two LIs have matching licensor-licensee features as their first features, often written as polar pairs (e.g., +f, -f). This is illustrated in (3).

(3)



In (3), after other merge features are checked, the T head and the subject DP have matching `k` features as their first features. Movement is licensed. In contrast to a phrase structure tree where the mover is indicated at its landing site, the subject remains at its merge position in (3). Trees such as this are derivation trees. The central role derivation trees play in MGs and MG parsing is discussed in Graf et al. (2017). We will also use derivation trees as the data structure for our processing model.

A note on notation before proceeding. In the above derivation trees, double-colon (::) indicates a LI, while a single colon (:) indicates a derived category. Phrase node names are added wherever helpful for readability. For all subsequent trees, we will omit features, lexical/derived category distinctions, and use phrase names for tree nodes. Movement arrows will also be added when helpful.

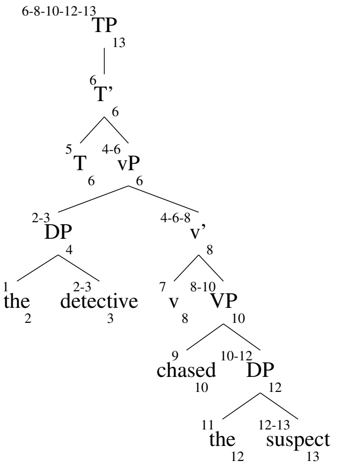
2.2 Left-corner MG parsing and complexity metrics

MG parsing can be viewed as a structural building process where a parser operates on MG rules, takes a string of words as input, and outputs a derivation tree when there is a valid parse. The left-corner parser for MGs used in this study is an arc-eager move-eager left-corner parser based on Stanojević and Stabler (2018); Hunter et al. (2019), in which the readers can find the full definitions of the parsing rules. For our purpose, we focus on tree annotations which are faithful visual representations of how the parser builds/traverses derivation trees.

Consider an arc-eager move-eager left-corner parse for the sentence (with silent nodes and string spans added) in (4). The parse history is represented using tree annotations in (5).

(4) $_{1}$ The $_{2}$ detective $_{3}$ T $_{3}$ v $_{3}$ chased $_{4}$ the $_{5}$ suspect $_{6}$

(5)



Following conventions in top-down MG parsing literature (e.g., Kobele et al. 2013; Graf et al. 2017), the superscripts and subscripts on the tree nodes, called indices and outdices, represent the steps at which that node enters and exits the memory storage of the parser. The dashes in the index of a node, which we use uniquely for left-corner parsing, connect the steps at which the parser updates its prediction regarding that node. Derivation trees annotated with indices, outdices, and dashes are shown to be condensed yet complete representations of the behavior of the left-corner MG parser (Liu, 2023, under revision). Building on this, we focus on the parser's updates represented with the dashes in the indices and show how to build complexity metrics based on them.

The update can be understood by examining the correspondence between parse items and derivation tree fragments. One node in the derivation tree can correspond to multiple strictly different parse items for a left-corner MG parse. For example, in (5) the parser reads the first input word *the* (step 1) and makes a left-corner prediction based on it (step 2), creating a parse item which takes the form of an implication shown in (6).

(6) `(2-i) n, M => (1-i) d, M`

This parse item is interpreted as follows, if from the string span of `(2-i)` the parser finds an item with category feature `n` and an optional mover chain `M`, the parser can infer that from the string span of `(1-i)` there is an item of category `d` which carries over the mover chain `M`. In terms of tree fragments, (6) corresponds to a DP with a daughter node yet to be confirmed. This is also the tree portion annotated with indices and outdices up to 2, matching the steps so far.

Next, when the parser reads *detective* from the input (step 3), the left-hand side of the implication in (6) is satisfied, a new parse item (7) is created at the same step and replaces (6).

(7) `(1-2) d`

This parse item means that from the string span of `(1-2)`, there is an item of category `d` without any mover chain. In terms of tree fragments, (7) corresponds to the fully built DP *the detective*. At step 3, both daughters of the DP are fully annotated. The DP node itself has an index of 3 and no outdex, meaning that it is still in memory at this step, ready for further operations.

Both the right-hand side in (6) and the whole item in (7) correspond to the same DP node in the derivation tree. The parser updates its knowledge of the node from a conditioned inference to a confirmed node. And the dashed index on the DP node records the steps at which the parser makes those updates. By taking the difference between the two dash-connected steps, we get the number of steps a parse item needs to be stored in memory, or its *item tenure*. For example, the parse item in (6) has a trivial item tenure of 1, as it is only stored between steps 2 and 3.

For a non-trivial example, vP has in its index 4-6. The parser first updates its knowledge on the vP node when it makes a left-corner prediction based on the DP *the detective*. A vP with a daughter node yet to be confirmed is created and held in memory. The parser's second update happens after the T head is read and processed. The time between the two updates is recorded with the dash-connected step pair. By taking the difference of the pair, we have the item tenure of the partially built vP, 2.

Item tenure serves as the basis for the complexity metrics of our left-corner MG parsing model. There are many ways to construct complexity metrics based on item tenure. Liu (under revision) explores a few of those possibilities. Here we focus on Maximal item tenure (MaxT$_{item}$) and its recursive variant (MaxT$^{R}_{item}$). MaxT$_{item}$ is the maximal duration that any parse item remains in memory. MaxT$^{R}_{item}$, following Graf et al. (2017), applies MaxT$_{item}$ recursively. MaxT$_{item}$ is shown to be able to capture the processing of sentence embeddings (Liu, 2024), it is included here to further test its reliability. In cases of a lack of word order preferences, we expect to find a tie in MaxT$_{item}$ for the word order pair. Examining MaxT$^{R}_{item}$ in those cases helps reveal further potential processing differences.

With methods and tools ready, we turn to the modeling results.

3 Modeling results

The processing phenomena modeled with the left-corner MG parser are the short-before-long preference in the English heavy NP shift (HNPS); the long-before-short preference in the Japanese transitive sentences; and the absence of word order preference in preverbal PPs in Mandarin Chinese. For each case, we make pairwise comparisons between the two opposite word orders (e.g., shift vs. canonical word order in English heavy NP sentences).

Overall, MaxT$_{item}$ successfully captures all three word order preferences. The preferred order has a lower MaxT$_{item}$ in both the English (short-before-long) and Japanese (long-before-short) target sentences. Furthermore, MaxT$_{item}$ predicts a tie in processing difficulties in the Mandarin (no preference) sentences. Since our goal is to understand the interplay of specific syntactic structures and a general memory constraint on processing, we next examine the structural assumptions and the complexity metric in each word order pair.

3.1 Short-before-long preference

The target sentences for the short-before-long preference are the canonical (8) and heavy NP shift order (9) in English (with silent heads).

(8) Max T v-put all the boxes of home furnishings V in a car.

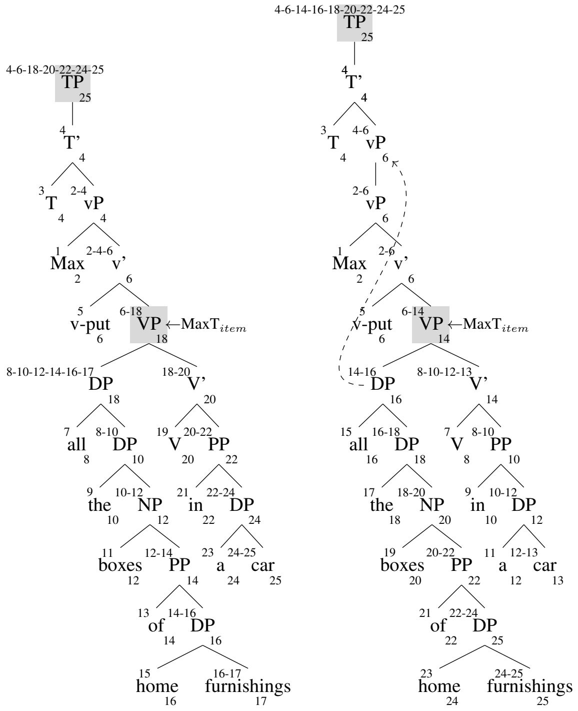
(9) Max T v-put V in a car all the boxes of home furnishings.

Evidence for the short-before-long preference in the above sentences is found in numerous behavioral and corpus studies (e.g., behavioral: Stallings et al. 1998; Stallings and MacDonald 2011; corpus: Wasow 2002; Liu 2020). For our model, we expect to find that the shifted order has a lower MaxT$_{item}$ compared with that of the canonical order, suggesting that the former is easier to process.

In terms of structural assumptions, a rightward movement analysis (Ross, 1986; Overfelt, 2015) is adopted to derive the heavy NP shift order. V-to-v and AgrO movements are factored out for simplicity.

The modeling results suggest that the shift order is easier to process than the canonical order. MaxT$_{item}$ for the shift order is 12 compared with 8 for the canonical order. The reason for the difference in MaxT$_{item}$ can be seen from the tree annotations in Figure 1.

For both word orders, the MaxT$_{item}$ is associated with the VP node. As the parser processes the verb *v-put*, a left-corner prediction based on the node predicts and stores an implicational parse item involving VP: if the parser finds a VP, it can confirm that there is a TP. Given the arc-eager strategy, this stored VP node is considered found when the parser makes a left-corner prediction based on one of its fully built daughter. And this is when word order makes a difference. If the parser first builds the less complex daughter, the V', the VP is held in memory for less time than when building the more complex daughter first. This is reflected in the difference in MaxT$_{item}$, as can be seen in Figure 1a for the canonical order and Figure 1b for the shift order.



(a) HNPS - Canonical order (b) HNPS - Shift order

Figure 1: Tree annotations for short-before-long preference

This is an encouraging result as it indicates that the left-corner MG parsing is at least as good as its top-down variant in capturing the short-before-long preference. We now turn to the long-before-short preference, where the top-down model struggles.

3.2 Long-before-short preference

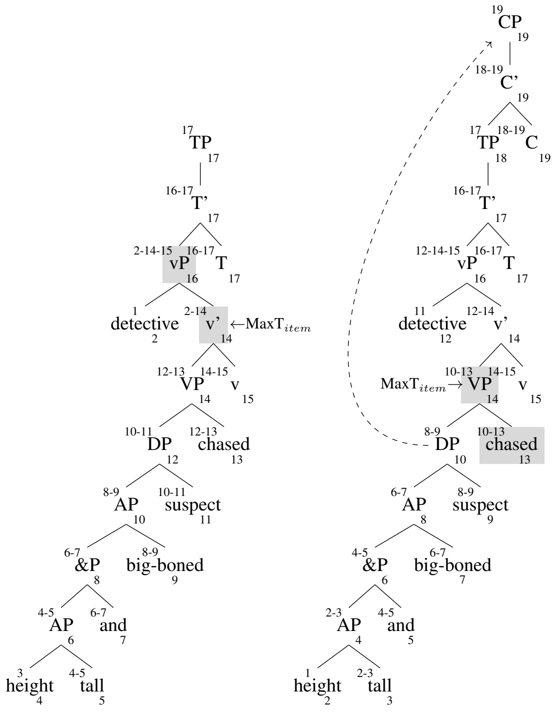
The long-before-short preference we model is reported in Yamashita and Chang (2001) regarding Japanese transitive sentences. The study finds that in a sentence production task, Japanese-speaking participants tend to order long arguments ahead of short ones. For example, compared with a canonical SOV order in (10), a long-before-short OSV order in (11) is preferred when the object is long.

(10) keezi-ga Se-ga takakute
detective-nom height-nom tall-and
gassiri sita hanni-o oikaketa v T
big-boned suspect-acc chased

(11) Se-ga takakute gassiri sita hanni-o
height-nom tall-and big-boned suspect-acc
keezi-ga oikaketa v T C
detective-nom chased
'The detective chased the suspect who is tall and big-boned.'
(adapted from Yamashita and Chang 2001, silent nodes added)

The sentence pair in (10-11) is used in our model as target sentences. A scrambling analysis is assumed to derive the long-before-short order (Saito, 1992). V-to-v and AgrO movements are again factored out for simplicity.

The modeling results show that the shift, long-before-short word order is easier to process than the canonical order. MaxT_{item} of the shift order is 3 compared with 12 of the canonical order. The tree annotations confirm the processing prediction.



(a) Japanese - SOV order (b) Japanese - OSV order

Figure 2: Tree annotations for long-before-short preference

In Figure 2a which corresponds to the canonical order, MaxT_{item} is associated with the v' node. The parser predicts and stores a parse item with v' when the subject, *detective*, is processed. The parse item is flushed from memory when one of the daughters of v' is built and used for left-corner prediction. Given the word order, this only happens after the long DP (indeed, the full VP) is fully built, resulting in large item tenure. In the long-before-short tree in Figure 2b, the parser builds the long DP first, during which process no other parse item is held in memory. As a result, item tenures and MaxT_{item} stay relatively low throughout the parse, predicting that the long-before-short order is easier to process than the canonical order.

3.3 Absence of order preference

Liu (2020) reports in a large-scale corpus that Mandarin Chinese preverbal PPs lack a preference for word order when the two PPs are of different lengths. For example, no word order preference is found between whether ordering the longer PP first (12) or the shorter first (13).

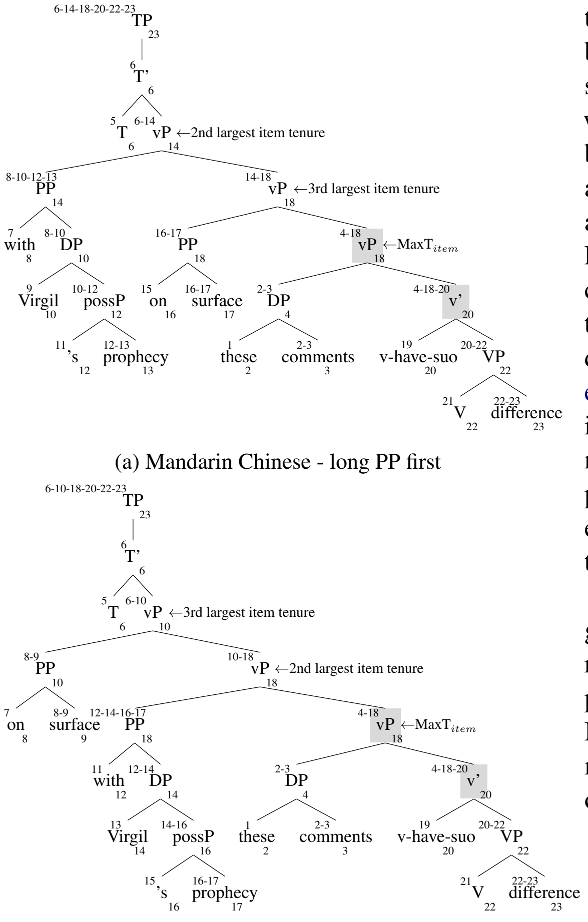
(12) zhexie yanlun T [he weijier de yuyan]
these comments with Virgil's prophecy
[zai biaomian] v-you-suo V churu
on the surface have-suo differences

(13) zhexie yanlun T [zai biaomian]
these comments on the surface
[he weijier de yuyan] v-you-suo V
with Virgil's prophecy have-suo
churu
differences
'These comments have differences on the surface with Virgil's prophecy.'
(from Liu 2020, silent nodes added)

(12) and (13) are the target sentences to include in our model. In terms of the structural assumption, the two PPs are considered based-generated adjuncts. Similar to before, V-to-v and AgrO movements are factored out for simplicity. Unlike before, the two word orders are not derivationally related under the current structural assumption. We will consider an alternative analysis in the context of methodological discussion in Section 4.

The results show that the two orders are indistinguishable for our model based on MaxT_{item}. MaxT_{item} is 14 for both orders, suggesting that no preference is expected for the two word orders. We see why MaxT_{item} is unaffected by word order alternations in the tree annotations in Figure 3.

Given the current structural assumption, MaxT_{item} is associated with the vP node immediately dominates the subject *these comments*. The parser creates and stores a parse item with this vP node when the subject is processed. This parse item is flushed from memory after the inner PP, or the linearly second PP, is processed. Alternating the order of the two PPs would not affect the item



(a) Mandarin Chinese - long PP first

(b) Mandarin Chinese - short PP first

Figure 3: Tree annotations for Mandarin Preverbal PPs

tenure of the parse item with the vP node created early on.

Interestingly, MaxT$^{R}_{item}$, a recursive evaluation of MaxT$_{item}$, also predicts that there is no preference between the two orders. In the two orders, the second largest item tenures are equal, so are the third largest. They are associated with the mother node of the longer and the short PPs respectively. Because of the structural similarity, all other item tenures are equal, too. An alternation of word order does not affect the item tenure profile.

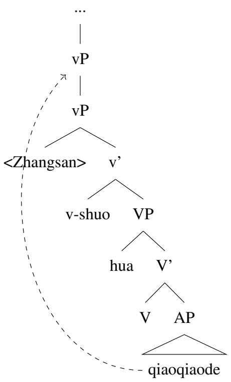
4 Discussions: an alternative structure for Mandarin adjuncts

The modeling results have shown that left-corner MG parsing is an effective model for word order preferences crosslinguistically. MaxT$_{item}$ has proven to be a reliable complexity metric capturing the mixed word order preferences under the current syntactic assumptions. Among those assumptions, the base-generation analysis of Mandarin preverbal PPs warrants particular attention. While it is standard to treat PP adjunction as base-generation, with word order alternation derived from different base merge positions, the choice of this structural assumption has a potential limitation: it can be adequately captured by a Context-Free Grammar. For both formalisms, no movement is involved that causes a mismatch between the string order and the leaf order. The ability to handle this mismatch distinguishes MG parsers from CFG parsers (Graf et al., 2017). As a result, for our purposes, processing models based on this syntactic assumption may not fully highlight the unique contribution of MG parsing in capturing the interplay between general efficiency principles and detailed syntactic structures.

Furthermore, there are syntactic proposals regarding other types of adjuncts in Mandarin that require the expressive power of MGs. For example, (Larson, 2018) argues that manner adverbs in Mandarin Chinese merge as VP complement and move to vP edge which derives the correct word order. This is schematized in (14).

(14) a. Zhangsan qiaoqiaode shuo hua
Z quiet-de speak words
'Z. speaks quietly.' (Larson, 2018)

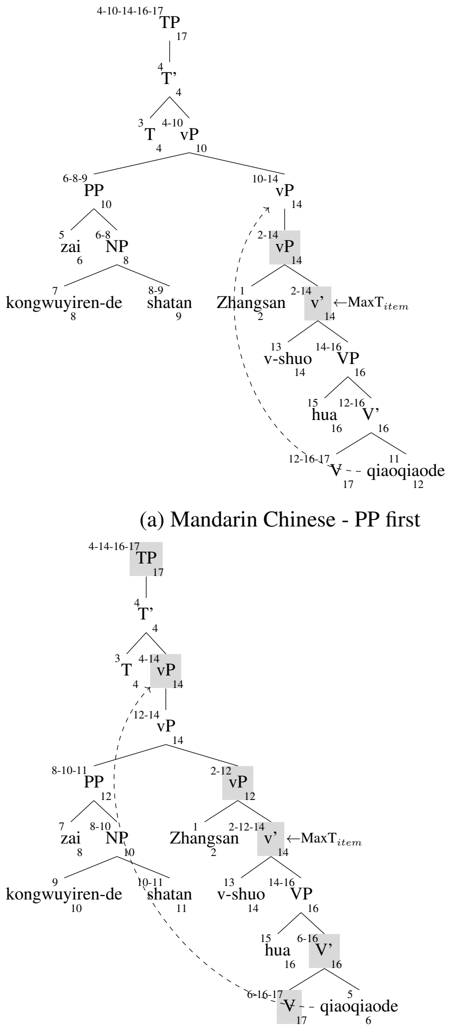
b.



We next model how this syntactic proposal affects order preferences. The target sentences (with silent heads) are shown in (15) and (16) corresponding to the PP-first and adverb-first order, respectively.

(15) Zhangsan T zai kongwuyiren de shatan
Z. at not-a-single-person de beach
qiaoqiaode v-shuo hua V
quite-de speak word

(16) Zhangsan T qiaoqiaode zai
Z. quite-de at
kongwuyiren de shatan v-shuo hua V
not-a-single-person de beach speak word



(a) Mandarin Chinese - PP first

(b) Mandarin Chinese - adverb first

Figure 4: Tree annotations for Mandarin PP and AP adjuncts

'Z. speaks quietly at an empty beach.'

For syntactic assumptions, the manner adverb is analyzed according to Larson (2018). The PP adjunct is base-generated either before or after the manner adverb moves to derive the two word orders. This is illustrated with annotated derivation trees in Figure 4.

The modeling result suggests that an AP-first order is preferred irrespective of the length of the two phrases. In both word orders, MaxT$_{item}$ is associated with the mother and sister node of the subject *Zhangsan*. The parse item associated with the two nodes is stored until the parser updates its knowledge on either node. For both orders, this happens after the parser has processed the AP and the PP. This means the lengths of the two phrases have the same effect on MaxT$_{item}$ for both orders. In the PP-first case in Figure 4a, it is the v' node that gets an update as the parser processes the two adjuncts and the verb *v-shuo*. In the AP-first case in Figure 4b, the vP node gets an update as soon as the two adjuncts are built and processed. This results in a constant MaxT$_{item}$ advantage of 2 (10 vs. 12) for the AP-first order over the PP-first order.

The result does not immediately rule out the possibility that there is no preference for ordering shorter or longer phrases first. Empirical data is needed to verify whether there is a preference for AP-first ordering and to assess its implications for the DLM principle. We leave these intriguing questions for future research.

5 Conclusion

This paper offered a unified, structure-based account of crosslinguistic word order preferences using Minimalist Grammars and a left-corner MG parser. The results show that preferred word orders correspond to structures that are less memory-intensive to process, and that no memory load difference is observed—given the current complexity metric—in cases that lack a word order preference. This supports the view that word order preferences follow from syntactic structure and highlights the potential of left-corner MG parsing as a psycholinguistically grounded model of sentence processing.

References

Noam Chomsky. 2014. *The minimalist program*. MIT press.

Aniello De Santo. 2019. Testing a minimalist grammar parser on italian relative clause asymmetries. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 93–104.

Aniello De Santo. 2020. *Structure and memory: A computational model of storage, gradience, and priming*. Ph.D. thesis, State University of New York at Stony Brook.

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

Edward Gibson. 2000. The dependency locality theory: A distance-based theory of linguistic complexity. *Image, language, brain*, 2000:95–126.

Thomas Graf, James Monette, and Chong Zhang. 2017. Relative clauses as a benchmark for minimalist parsing. *Journal of Language Modelling*, 5(1):57–106.

John A Hawkins. 1994. *A performance theory of order and constituency*, volume 73. Cambridge University Press.

John A Hawkins. 2004. *Efficiency and complexity in grammars*. Oxford University Press on Demand.

Tim Hunter, Miloš Stanojević, and Edward Stabler. 2019. The active-filler strategy in a move-eager left-corner minimalist grammar parser. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 1–10.

Gregory M Kobele, Sabrina Gerth, and John Hale. 2013. Memory resource allocation in top-down minimalist parsing. In *Formal Grammar*, pages 32–51. Springer.

Richard K Larson. 2018. Ap-de adverbs in mandarin. *Studies in Chinese Linguistics*, 39(1):1–28.

So Young Lee. 2018. A minimalist parsing account of attachment ambiguity in english and korean. *Journal of Cognitive Science*, 19(3):291–329.

Haitao Liu. 2008. Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science*, 9(2):159–191.

Lei Liu. 2022. *Phrasal Weight Effect on Word Order*. Ph.D. thesis, State University of New York at Stony Brook.

Lei Liu. 2023. Processing advantages of end-weight. *Proceedings of the Society for Computation in Linguistics*, 6(1):250–258.

Lei Liu. 2024. Psycholinguistic adequacy of left-corner parsing for minimalist grammars. *Proceedings of the Society for Computation in Linguistics (SCiL)*, pages 275–280.

Lei Liu. under revision. Psycholinguistic plausibility of left-corner parsing with Minimalist Grammars.

Zoey Liu. 2020. Mixed evidence for crosslinguistic dependency length minimization. *STUF-Language Typology and Universals*, 73(4):605–633.

Jason Overfelt. 2015. Rightward movement: A study in locality.

John Robert Ross. 1986. *Infinite syntax*. Ablex Publishing Corporation.

Mamoru Saito. 1992. Long distance scrambling in japanese. *Journal of East Asian Linguistics*, 1(1):69–118.

Edward Stabler. 1997. Derivational minimalism. In *Logical Aspects of Computational Linguistics: First International Conference, LACL'96, Nancy, France, September 23-25, 1996. Selected Papers*, volume 1328, page 68. Springer Science & Business Media.

Edward P Stabler. 2011. Computational perspectives on minimalism. *Oxford handbook of linguistic minimalism*, pages 617–643.

Edward P Stabler. 2013. Two models of minimalist, incremental syntactic analysis. *Topics in cognitive science*, 5(3):611–633.

Lynne M Stallings and Maryellen C MacDonald. 2011. It's not just the "heavy np": relative phrase length modulates the production of heavy-np shift. *Journal of psycholinguistic research*, 40(3):177–187.

Lynne M Stallings, Maryellen C MacDonald, and Padraig G O'Seaghdha. 1998. Phrasal ordering constraints in sentence production: Phrase length and verb disposition in heavy-np shift. *Journal of Memory and Language*, 39(3):392–417.

Miloš Stanojević and Edward Stabler. 2018. A sound and complete left-corner parsing for minimalist grammars. In *Proceedings of the Eight Workshop on Cognitive Aspects of Computational Language Learning and Processing*, pages 65–74.

T Wasow. 2002. *Postverbal Behavior*. CSLI Lecture Notes (CSLI- CHUP) Series. CSLI Publications.

Hiroko Yamashita and Franklin Chang. 2001. "long before short" preference in the production of a head-final language. *Cognition*, 81(2):B45–B55.

Chong Zhang. 2017. *Stacked Relatives: Their Structure, Processing and Computation*. Ph.D. thesis, State University of New York at Stony Brook.

MGEN: Millions of Naturally Occurring Generics in Context

Gustavo Cilleruelo Calderón **Emily Allaway** **Barry Haddow** **Alexandra Birch**
School of Informatics, University of Edinburgh
g.cilleruelo-calderon@sms.ed.ac.uk {emily.allaway, bhaddow, a.birch}@ed.ac.uk

Abstract

MGEN is a dataset of over 4 million naturally occurring generic and quantified sentences extracted from diverse textual sources. Sentences in the dataset have long context documents, corresponding to websites and academic papers, and cover 11 different quantifiers. We analyze at scale the features of generic sentences, with interesting insights: generics can be long sentences (averaging over 16 words) and speakers often use them to express generalisations about people.

MGEN is the biggest and most diverse dataset of naturally occurring generic sentences, opening the door to large-scale computational research on genericity. It is publicly available at gustavocilleruelo.com/mgen.

1 Introduction

Generics are sentences that express generalisations without making use of explicit quantifiers. Examples of generics are *ravens are black* or *ticks carry lyme disease*.

Several features of generics make them difficult to account for semantically (Carlson and Pelletier, 1995): they are permissive to exceptions (*ravens are black* is acceptable even if albino ravens exist) and the quantifications they convey have paradoxical dynamics (Leslie, 2008). If we paraphrase the previous generics as explicitly quantified, we would have *most ravens are black* but *few ticks carry lyme disease*: the same linguistic structure conveys generalisations at opposite ends of the quantification spectrum.

In this work, we introduce MGEN, a dataset designed to provide a solid foundation for research on generic sentences in English. MGEN has 4.1 million samples, with over 3 million generics and 1 million explicitly quantified sentences with 11 different quantifiers. All sentences are naturally occurring and include the context document in which they originally appear.

To motivate the design of MGEN, we conduct an extensive review of datasets of generic sentences and argue that existing datasets have many shortfalls: they are either small, rely on synthetic samples or have no context, despite theoretical works showing the importance of context for the semantics of generics (Sterken, 2015; Almotahari, 2023).

In order to mine generic sentences from massive corpora, we introduce a two-step pipeline: a syntactic filter detects bare plurals (this is the most common syntax of the subject for generics, see §2) with the required verb features and then a binary classifier labels them as generic or not. We apply this pipeline to a subset of the ZYDA (Tokpanov et al., 2024) dataset (a language model pre-training corpus) to collect a diverse and accurate (as per human annotators) dataset of generic sentences.

We analyze the corpus-level characteristics of MGEN and find that its generic sentences are longer than those usually considered in the literature, where running examples are much shorter than the average 16.65 words in our dataset. Analysing the word frequencies of our dataset, we find that speakers use generics most often to generalize about *people*.

Our contributions are: (i) MGEN, the largest dataset of naturally occurring generics in context, (ii) a pipeline for the extraction bare plural generics from textual sources, (iii) a review of existing datasets of generics and (iv) a preliminary corpus-level analysis of the characteristics of generic sentences.

2 Background: generics & quantifiers

Generics have kind terms in their subject position (i.e. words or phrases used to categorize or label groups of entities) and their verbs are inflected for third person plural present indicative. They are used either to make claims about those kinds (*dinosaurs are extinct*) or to attribute properties to

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 244-260.
Eugene, Oregon, July 18-20, 2025

Source	Sentence
RefinedWeb	Soybeans contain an inhibitor of trypsin, an enzyme important for digestion, but it can be destroyed by cooking.
SlimPajama	Cucumbers are high in an antioxidant called beta-carotene, which your body turns into vitamin A. May ease muscle cramps.
The Pile	Starving people grab the bread first and run with it.
arXiv	Colexification networks encode affective meaning.
peS2o	Car seats save lives.

Table 1: Examples of generic sentences from the different sources of MGen. More examples in Appendix F.

individuals in those kinds (*beetles have protective wing covers*).

Following most of the linguistics and philosophy of language literature, we consider only *bare plural* generics (Carlson and Pelletier, 1995; Leslie, 2007a). Bare plurals have noun phrases in plural form without a definite or indefinite article[1]. Throughout the paper, we will use *bare plural sentence* to refer to sentences with the syntax of a bare plural generic (i.e. with the same inflection of the verb), even if those sentences are not generics.

The standard view in linguistics is that generics are quantificational: there is an unpronounced operator GEN that takes a role similar to adverbial quantifiers in the logical form of the sentence (Lewis, 1975; Carlson, 1977b; Carlson and Pelletier, 1995; Cohen, 1999b; Kirkpatrick, 2024).

In contrast, recent influential accounts of generics have been non-quantificational: Leslie (2008) gives generics the privileged role of expressing default or primitive generalisations, Sterken (2015) argues that quantification cannot capture the full context-sensitivity of generics and Nickel (2016) relates generics to a notion of normality grounded in explanatory considerations rather than the prevalence of the property in the kind.

The rich landscape of theories of generics, as well as their far-reaching implications into fundamental aspects of human cognition, has made the study of generic sentences a highly debated topic in recent years (e.g., Cohen, 1999a; Tessler and Goodman, 2016; Stovall, 2019; Nguyen, 2020; Bosse, 2021; Almotahari, 2022; Kirkpatrick, 2023; Neufeld et al., 2025)

In the field of natural language processing, recent works study how language models deal with aspects of genericity such as exceptions, property inheritance (Allaway et al., 2024) and quantification (Ralethe and Buys, 2022; Collacciani et al., 2024). Cilleruelo et al. (2025) uses language models to study the semantics of generic sentences, such as their implicit quantification.

3 Related work: datasets of generics

Several datasets exist that specifically target generics. We compare these datasets across four dimensions (Table 2): total samples, quantified sentences, context and origin (natural or synthetic).

We consider *natural* sentences to be only those that have been extracted from human-written sources and *synthetic* those have been either generated by language models, built with rule-based methods or constructed by researchers or annotators. We also include quantified sentences as a requirement for datasets of generics as these are a key contrast class. Similarly, context plays an important role on the semantics of generics.

GenericsKB (Bhakthavatsalam et al., 2020) is a dataset that is composed of both naturally occurring generic and quantified sentences in context and synthetic examples derived from knowledge bases.

To source the naturally occurring samples, $3.5M$ candidate sentences are extracted from different corpora (Wikipedia, ARC and Waterloo) through 27 hand-crafted lexico-semantic rules. A subset of those are manually annotated and used to train a Bert-based binary classifier (generic and not generic).

This classifier is used to score the $3.5M$ candidate sentences to curate GenericsKB-Best: a collection of the best-scoring naturally occurring sentences ($N = 774,621$) augmented with synthetic generics derived from knowledge bases ($N = 246,247$). Some sentences are quantified with *all*, *most*, *some*, *many*, *every*, *much*, *more*, *often*, *usually*, *always*, *sometimes*, *frequently*.

[1] *Tigers have stripes* is a bare plural generic, which can also be expressed in English with the definite (*the tiger has stripes*) or indefinite (*a tiger has stripes*) articles.

Dataset	Scale	Quantifiers	Context	Sources
MGEN (Ours)	$4.1M$	Yes (11)	Yes	Natural (ZYDA)
GENERICSKB-BEST (Bhakthavatsalam et al., 2020)	$1M$	Yes (13)	Yes	Natural (Waterloo, SimpleWiki, ARC) Synthetic (WordNet, ConceptNet, TupleKB)
CONGEN (Cilleruelo et al., 2025)	2872	Yes (3)	Yes	Natural (DOLMA)
GEN-A-TOMIC (Bhagavatula et al., 2023)	$> 8M$	Yes (3)	No	Synthetic (GPT2-XL with I2D2)
Animal generics (Ralethe and Buys, 2022)	75,002	No	No	Mixed (GENERICSKB)
EXEMPLARS (generics) (Allaway et al., 2024)	$16,655$	No	No	Mixed (GEN-A-TOMIC, Animal generics)
Dataset in (Collacciani et al., 2024)	1837	Yes (5)	No	Synthetic (human annotations)
Norwegian generics (Kurek-Przybilski and Adam, 2022)	170	No	Yes	Natural (encyclopedia entries)

Table 2: Comparison between existing datasets of generic sentences. MGEN is comparable in size with synthetic datasets but is comprised of naturally occurring sentences in context.

Cilleruelo et al. (2025) introduce CONGEN, a collection of 2873 naturally occurring generic and quantified sentences in context. Because the dataset is manually curated, it is small and only contains data for 3 quantifiers (*all*, *most* and *some*).

The biggest dataset of synthetic generics is the GEN-A-TOMIC corpus (Bhagavatula et al., 2023). Sentences in GEN-A-TOMIC are generated by GPT2-XL (Radford et al., 2019) through knowledge distillation with self-imitation algorithm. Although GEN-A-TOMIC has over 8 million utterances, because they are generated with a small language model, these are not in context and the only quantifiers included are *generally*, *typically* and *usually*.

Ralethe and Buys (2022) select generics and quantified sentences from GENERICSKB by filtering for animals, curating a subset of 75,002 generics. This collection of animal generics is combined with examples from GEN-A-TOMIC to create datasets of synthetic generics exemplars (i.e. cases where the generic does and does not hold) (Allaway et al., 2023, 2024), which contain generic sentences, as well as their derived exemplars.

To conduct experiments on language models, Collacciani et al. (2024) collect 1873 sentences from three sources, all crafted either by researchers or annotators (Herbelot and Vecchi, 2016; Urbach and Kutas, 2010; Misra et al., 2023). Sentences in this dataset are extremely short (average length is 3.73 ± 1.03, median is 3) and all are annotated with a quantifier (*all*, *most*, *some*, *few*, *no*).

All datasets considered so far, as well as MGEN, are in English. In Norweigan, Kurek-Przybilski and Adam (2022) manually extract 170 generics in context from encyclopedic texts.

Table 2 compares the reviewed datasets of generic sentences in terms of total samples, inclusion of quantified sentences, context for the utterances and data origin. Our dataset, MGEN, has the scale of GENERICSKB and GEN-A-TOMIC, but without the need of synthetic examples (whether generated or constructed from knowledge bases) and includes context documents for all generic as well as quantified utterances.

4 Methodology

This section details the construction of the MGEN dataset. We first describe the high-level objectives for the creation of the dataset, based on the generics literature and the shortcomings of existing datasets. Then, we detail the extraction of generics and quantified sentences at scale from a large corpus by leveraging syntactic (§4.4) and semantic (§4.5) characteristics of generics.

4.1 Design choices

MGEN is built to include a massive, diverse amount of naturally occurring generic sentences with their respective contexts. In this section we go over the principles that guide the construction of the dataset.

Naturally occurring. We focus on naturally occurring generic sentences, as it would be hard to assess the acceptability of synthetic samples without assuming a theory of generics or conducting extensive human annotation studies, since the semantics of generics are not well understood (§2).

Context. Many works argue that the context radically affects what generic sentences express, for example, in terms of both quantificational strength and flavor (Sterken, 2015; Almotahari, 2023). To mine generic sentences, we choose a corpus structured in documents (more details in §4.2) and keep the full context document of each sample.

Bare plurals. We focus on generics that are bare plurals (§2) and only at the beginning of a sentence. This makes detection at scale more tractable, by, for example, omitting nested generics in *that* clauses (e.g. *she maintains that the belief that technology improves education is widely accepted*).

Quantifiers. Generics and quantified sentences are closely related, as both are used to express generalisations. We collect quantified sentences with the following structures: *quantifier* + *bare plural sentence*, *bare plural noun phrase* + *quantifier* + *verb* or *bare plural noun phrase* + *verb* + *quantifier*. We consider the following 11 quantifiers: *all*, *most*, *many*, *some*, *few*, *no*, *often*, *generally*, *typically*, *usually*, *normally*.

4.2 Data sources

Training language models requires large collections of clean textual data, which can also be used for data mining. We use ZYDA (Tokpanov et al., 2024), an open-source dataset built by collecting text from different high-quality sources and performing uniform filtering and deduplication. We run our generic extraction pipeline on the following components of ZYDA (Appendix E; Table E.3): RefinedWeb (Penedo et al., 2023), SlimPajama (Soboleva et al., 2023), the Pile (Gao et al., 2021), peS2o (Soldaini and Lo, 2023) and arXiv (Kenney, 2023).

RefinedWeb, SlimPajama and The Pile primarily consist of data scraped from the web, while the much smaller peS2o and arXiv are composed of academic publications.

4.3 Generic sentence extraction

ZYDA is structured in documents: roughly the text in a website, a scientific article or similar. Each document is first split into sentences (`blingfire`[2]). Then, a lightweight syntactic filtering step selects sentences where either (i) the first word is one of the quantifiers of interest, or (ii) there is a *plural noun* in the first 4 words of the sentence (`flair` (Akbik et al., 2019)).

These candidates are then run through two filtering steps: a syntactic one that ensures these are bare plurals with verbs inflected for third person present indicative and a semantic one, that filters for sentences that express generalizations. This latter step is necessary as the bare plural generic syntactic construction can also have existential readings, where the subject refers to specific instances instead of to a kind in general, e.g. *tigers are in the front lawn* or *blue arrows indicate acceleration* (also see Appendix F; Table F.6).

We detail the construction of each filtering step in §4.4 and §4.5 respectively.

4.4 Syntactic filtering (bare plurals)

The syntactic filtering step in the pipeline receives candidate sentences with plural nouns in the early words and performs a more in-depth dependency analysis to select only bare plural sentences.

The part-of-speech and dependency parsing of the sentence is conducted with the `stanza` python library (Qi et al., 2020). After parsing the sentences, we keep those that meet the following three conditions:

1. The nominal subject is a plural noun or a plural proper noun (`nsubj` or `nsubj:pass` in the case of passives).

2. The root of the nominal subject is a verb or an auxiliary (`VERB` or `AUX`). If there is a copula (`cop`) or a passive (`aux:pass`), take that as the verb.

3. The verb has present tense, indicative mood, plural number and third person.

4.5 Semantic filtering (genericity)

The syntactic filtering step yields bare plural candidate sentences, but these include noisy and non-generic samples. To get high quality generics from these candidates, we apply a further step in which a binary classifier scores whether the bare plurals are generic or not.

This classifier is designed to filter out: (i) sentences that although they may contain a generic it is not at the beginning[3], (ii) sentences that are

[2] https://github.com/microsoft/BlingFire

[3] A common occurrence are titles of paragraphs or sections that get parsed at the beginning of the sentence, for example: *Gaussian Mixture Models Gaussian mixture models are*

ungrammatical or noisy and (*iii*) bare plurals that have existential (non-generic) readings (Table F.6).

We use a RoBERTa model (Liu et al., 2019) as the architecture for the classifer, which we train on a small collection of generics and non-generic bare plurals. The generics are sampled from GenericsKB-Best and the non-generics are generated by GPT-4 (OpenAI et al., 2024), by iteratively finding missclassified examples to make the training data more robust. The classifier achieves over 0.97 F-1 score in a test set based on ConGen and synthetic non-generic bare plurals. More details on classifier training and evaluation are found in Appendix A.

In the case of sentences that start with a quantifier, which are not bare plurals and are outside of the training distribution of the generics classifier, we remove the quantifier word and calculate the score of the resulting bare plural. This ensures that we pick out quantified sentences that are comparable to generics in terms of being generalizations as opposed to existential. We want to keep in the dataset sentences like *all tigers have stripes* but not *all tigers in the cage are male*.

Some quantified sentences begin with a bare plural rather than a quantifier (e.g. *tigers are normally striped*). For these sentences, we check if there is an adverbial quantifier that has as syntactic head the root of the sentence, and label them with the corresponding quantifier (if the quantifier is not in the main clause, the sentence is labeled as generic).

We include sentences that receive a genericity classifier score 0.8 or greater for the MGen dataset. This value is chosen by manual inspection of the data. The full unfiltered bare plurals data is also made publicly available.

5 MGen: Statistics & Analysis

In this section we summarize the statistics of the MGen dataset (§5.1) and present two quality analyses: human annotation to asses the genericity of the collected sentences (§5.2) and a comparison in terms of diversity with existing datasets (§5.3).

5.1 Statistics

We mine generics from a total of 50,534,844 Zyda documents (23% of the corpus). After the syntactic filtering of sentences for bare plurals, we end up with 16,771,049 sentences, of which **4,146,760** make up the final MGen dataset after receiving a score of 0.8 or higher by the generics classifier.

	Candidates	**Generalizations**
gen	14,303,840	3,183,293
All	502,629	82,752
Most	332,698	173,021
Many	389,606	188,419
Some	547,308	225,171
Few	22,164	8,085
No	47,146	4,121
Generally	116,901	53,015
Typically	124,522	53,046
Often	253,306	107,926
Usually	138,207	59,148
Normally	19,969	8,763
TOTAL	16,771,049	4,146,760

Table 3: Number of generics and quantified sentences after syntactic (candidates) and semantic (generalizations) filtering during the construction of MGen.

Source composition. The final dataset contains over 3 million sentences from internet crawls (RefinedWeb, The Pile and SlimPajama) and around 1 million sentences from academic sources, peS2o and arXiv (Appendix E; Table E.4). Of the total 4.1 million samples, about 3 million are bare plural generics, while the rest is made up of the 11 quantifiers in different proportions (Table 3).

Context documents. For every sentence in MGen, we include the document from Zyda that contains it. These documents correspond to websites or papers and are generally long, averaging over 5000 words. For comparison, the context documents in the samples of GenericsKB-Best are much shorter, with an average of 147 words.

Sentence length. We compute the length of sentences in words by splitting sequences by whitespaces. Figure 1 compares sentence length distributions for the naturally occurring examples in GenericsKB-Best, the generic (not quantified) sentences in MGen and the lengths in a sample of 20,000 context documents from MGen (Figure 1).

Generic sentences in MGen have an average of 16.65 ± 8.2 words and a median of 15 words: generics are often long sentences. Although generics are on average shorter than arbitrary sentences from MGen documents, the length distribution contrasts with the prototypical examples in the linguistics and philosophy literature, as well as many synthetic examples in computational linguistics, that usually have less than 5 words (for example,

formed by combining multivariate normal .... Note how the title (*Gaussian Mixture Models*) makes it so that the generic is not at the beginning.

Text	**Label 1**	**Label 2**	**Score**
Puppets are fun to include too.	Particular	Unclear	0.86
First thoughts are proverbially the best; at all events, they are the bravest.	Unclear	Generic	0.96
Pumps are used to circulate the water through collectors and into your water tanks.	Particular	Generic	0.97
Players get sets by asking another player for a specific card.	Generic	Particular	0.82

Table 4: Examples of annotator disagreements with classifier scores.

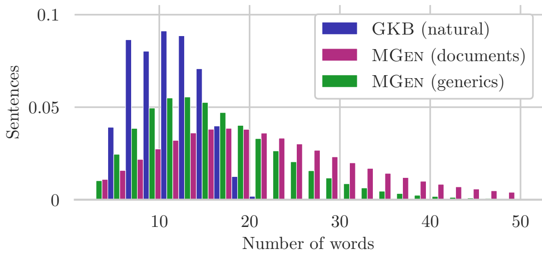


Figure 1: Sentence length distribution in the generics and documents of MGen and natural sentences in GenericsKB-Best.

see Appendix F, Table F.7 and examples in the Discussion §6). Examples of sentences in MGen with lengths from 3 to 25 words are available in Table F.9 (Appendix F).

Common words. The 50 most common words (excluding stopwords and punctuation) in MGen also reveal interesting aspects of the use of generics (Appendix E; Table E.5).

The most common word in MGen generics is *people*, with a big gap with respect to the second and third most common words: *also* and *cells*. In the generics of GenericsKB-Best, *also* is the most common word, and *water* and *one* are both more frequent than *people*, which is still fourth.

Following *people*, *women* and *children* are nouns with many occurrences, as well as terms specific to biology and medicine, such as *cells* and *patients*. The most common verb is *use* (and *used*, from passive constructions).

In contrast, we analyze the most common words in $100,000$ context documents from MGen and find that *people* does not even appear in the top 50: it is almost 60 times less prevalent $(16,5384)$ than the most common word, which is *also* with $942,208$ appearances.

These surface statistics of the sentences in the dataset give clues as to how we use generic sentences: to generalize about *people* and to express what to *use* things for.

In biology and medicine academic domains, which are well-represented in our dataset, we find a widespread use of generic sentences, as can be seen by the high frequency of some nouns particular to those fields.

5.2 Human evaluation of MGen

To evaluate the quality of samples in the MGen dataset in terms of genericty we use human annotators.

We sample 300 sentences from MGen which get annotated by two annotators by labeling the sentences as *Generic*, *Particular* (non-generic) or *Unclear*. Annotator guidelines are available in Appendix D. Examples with both annotations and the score of the RoBERTa classifier can be found in Table 4 and Table F.8 (Appendix F).

Annotators label 87.17% sentences as *Generic*, 7.5% as *Unclear* and 5.33% as *Particular*, with an 82% of inter-annotator agreement. Table 4 contains examples of disagreements. The human evaluation results suggest that, even as the annotation of generics is done automatically by a rather small model, the overall quality of the samples in MGen is high, making it a reliable source for generic sentences in context.

5.3 Diversity

We evaluate the diversity of the MGen dataset using three different measures: cosine similarity of sentence embeddings, distinct n-grams and distinct lemmas at subject, verb and object head positions.

Diversity from cosine similarity. Tevet and Berant (2021) introduce a transformation from pairwise sentence similarity to a diversity metric by taking an average of the similarity across possible sentence pairs (Eq. 1).

Given a corpus $\mathcal{C}$ and a 2-sentence similarity metric $m_{\text{sim}}(s_1, s_2) \in \mathbb{R}; s_1, s_2 \in \mathcal{C}$, the corre-

	diversity-from-similarity	distinct n-grams ($1M$ tokens)			head lemmas ($200k$ sentences)		
	$m_{\texttt{cossim}}$	distinct-1	distinct-2	distinct-3	Subject	Verb	Object
MGEN	$\mathbf{-7.09 \pm 0.13}$	$\mathbf{31,554}$	$\mathbf{396,923}$	$\mathbf{700,782}$	$\mathbf{18,836}$	$\mathbf{7,131}$	$\mathbf{15,935}$
GENERICSKB	-8.27 ± 0.14	$24,130$	$308,320$	$561,549$	$14,445$	$5,133$	$11,548$
GEN-A-TOMIC	-15.64 ± 0.2	$19,398$	$193,618$	$357,334$	$12,120$	$3,909$	$11,093$

Table 5: Diversity comparison of MGEN, GENERICSKB-BEST and GEN-A-TOMIC. In all scores higher is better.

sponding diversity-from-similarity metric as:

$$D_{\text{sim}}(\mathcal{C}) = -\frac{1}{\binom{|\mathcal{C}|}{2}} \sum_{s_i,s_j \in \mathcal{C}; i<j} m_{\text{sim}}(s_i, s_j) \quad (1)$$

We use as similarity function the cosine similarity ($m_{\texttt{cossim}}$) between sentence embeddings generated with NV-EMBED-V2 (Lee et al., 2024), a state-of-the-art model[4] in the Massive Text Embedding Benchmark (Muennighoff et al., 2023).

This diversity metric is computationally intractable for datasets with millions of sentences, we instead take 1000 samples of 1000 sentences each from the different datasets and report average diversity.

Diversity in distinct n-grams. We also consider an n-gram based diversity score, the distinct-n score (Li et al., 2015).

Given a corpus $\mathcal{C}$ with N_n n-grams and U_n unique n-grams. Then, the *distinct-n* score of $\mathcal{C}$ is the number of distinct n-grams (U_n) divided by the total number of words (N_1) in the corpus.

$$\text{distinct-}n_{\mathcal{C}} = \frac{U_n}{N_1} \quad (2)$$

We sample sentences from the each dataset until we reach 1 million tokens (as per the ROBERTA tokenizer). For clarity, we report the number of distinct n-grams directly, without normalizing by N_1, as all samples have the same size in total tokens.

Diversity from head lemmas. Because sentences in MGEN are naturally occurring, samples may have relative, subordinated or conjunctive clauses beyond the main bare plural generic, which could artificially inflate the n-gram count.

To have a fair comparison in this regard we introduce a score that counts the unique lemmatized verbs and head nouns in the subject and object positions. For each generic sentence, we get at most 3 lemmas, regardless of any clauses or subordinated sentences. For example, given *bees in the forests of Catalonia feed on lavender flowers, giving their honey a distinctive taste* would be reduced to 3 lemmas: *bee*, *feed* and *flower*. This way we target more directly the diversity in the generic sentences of the dataset.

We sample $200,000$ sentences from each dataset and report the total unique lemmas found.

MGEN is the most diverse generics dataset. We compare MGEN to GENERICSKB-BEST and GEN-A-TOMIC in terms of diversity by the three previous measures (Table 5). To make the comparison fair, we leave out synthetic samples from GENERICSKB-BEST, and use only the naturally occurring sentences.

In all cases, MGEN is more diverse than the comparable datasets of generics, both in lexical (distinct n-grams and head lemmas) and neural (cosine similarity) measures. This shows that the ROBERTA classifier, even if it is based on a relatively small model, is able to label a wide range of generics.

6 Discussion

In recent years, the study of generic sentences has focused on the careful consideration of a series of prototypical examples that highlight different aspects of their semantics. Some notable generics are *typhoons arise in this part of the Pacific* (Carlson, 1977b), *mosquitoes carry the West Nile virus* (Leslie, 2008), *ducks lay eggs* (Leslie et al., 2011), *humans kill themselves* (Sterken, 2015), *dobermans have floppy ears* (Nickel, 2016) and many others. Although these examples are effective at illustrating the semantics of generics, they are difficult to leverage computationally.

With the introduction of MGEN, a massive collection of naturally occurring generics in context, we open the door for new computational and corpus-level approaches to make progress in the puzzle of generics.

MGEN consists of 3 million generics and 1 million sentences explicitly quantified by 11 different quantifiers. These have been mined from a diverse pool of internet and academic documents, ensuring that many of the ways in which speakers use

[4] As of December 2024.

generics are represented.

Our analysis shows that MGEN is the more diverse of the large-scale datasets of generics, and human annotation suggests that, even as generics are automatically filtered, the quality of the examples is high.

If we take MGEN as a representative sample of generics, at least of some of the many ways in which English speakers use them, the statistics of the dataset say much about generics themselves.

The analysis of sentences in MGEN suggests that *generics are long*. They have over 16 words on average, with the most common sentence length being 15. Even if some generics in the dataset are long due to clauses and subordinate sentences, this still suggest sentences that begin with a generic express complex ideas. We also find many generics, in scientific and medical domains (Peters et al., 2024), that are not only long but contain many technical terms.

The technicality and length of many generics in MGEN contrasts with theories that link generics to "thinking-fast" or System I (Kahneman, 2011) in the dual-process theory of cognition (Leslie, 2007b; Almotahari, 2023). Combining the intuitive and unreflective use of generics, which speakers often do, with some of the long and complex sentences in MGEN is one of the open questions this dataset could help resolve.

We believe MGEN can play a role in future research on generics and quantifiers by providing examples with long context documents across a multiple sentence lengths (Appenix F; Table F.9) and topics, from academic papers to internet forums. These could disclose different ways in which speakers use generics. For example, that *people* is the most common noun suggests that generics play an important role on how humans understand each other through language.

7 Conclusion

In this work we build MGEN, a massive collection of generic and quantified sentences in context.

We mine generic sentences from ZYDA, a corpus for language model training. Our two-step pipeline first filters sentences by their syntactic features and then uses a ROBERTA-based classifier to determine genericity.

The final dataset contains over 3 million bare plural generics and 1 million quantified sentences with 11 different quantifiers. We believe MGEN is a valuable resource for future research on generic sentences.

The MGEN dataset is open-source, available at gustavocilleruelo.com/mgen.

Acknowledgements

This work was funded by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant number 10039436].

The first author would like to thank Mahrad Almotahari, Dan Lassiter, Nicolas Navarre and Aina Centelles Tarrés for their proof-reading corrections and discussions.

We also thank the anonymous reviewers for their insightful comments and feedback on the manuscript.

Limitations

Data contamination. This dataset is designed as a corpus for the study of language, rather than for any evaluation of the performance of language models. The sources that conform ZYDA are commonly used in the training of language models, which means any sort of performance evaluation in this data would be compromised and should be carefully carried out.

Generics classifier. The classifier that we use to classify generics as such does only take information from the sentence itself, we do not append any context. Future versions of the pipeline could use stronger models for selection of generics from bare plural sentences.

Distribution of generics. Although MGEN has millions of generics, it may not capture the full distribution of generic sentences: it only contains bare plural generics at the beginning of the sentence. Similarly, the quantified sentences we select are within a limited range of structures.

Three main assumptions underlie the generics of this dataset: (i) bare plurals (ii) at the beginning of the sentence (iii) in English. Future work that tries to capture more holistically generics across languages should improve upon these.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference* *of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Emily Allaway, Chandra Bhagavatula, Jena D. Hwang, Kathleen McKeown, and Sarah-Jane Leslie. 2024. Exceptions, Instantiations, and Overgeneralization: Insights into How Language Models Process Generics. *Computational Linguistics*, pages 1–60.

Emily Allaway, Jena D. Hwang, Chandra Bhagavatula, Kathleen McKeown, Doug Downey, and Yejin Choi. 2023. Penguins don't fly: Reasoning about generics through instantiations and exceptions. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2618–2635, Dubrovnik, Croatia. Association for Computational Linguistics.

Mahrad Almotahari. 2022. Weak generics. *Analysis*, 82(3):405–409.

Mahrad Almotahari. 2023. Generic cognition: A neglected source of context sensitivity. *Mind and Language*.

Chandra Bhagavatula, Jena D. Hwang, Doug Downey, Ronan Le Bras, Ximing Lu, Lianhui Qin, Keisuke Sakaguchi, Swabha Swayamdipta, Peter West, and Yejin Choi. 2023. I2d2: Inductive knowledge distillation with neurologic and self-imitation. *Preprint*, arXiv:2212.09246.

Sumithra Bhakthavatsalam, Chloe Anastasiades, and Peter Clark. 2020. Genericskb: A knowledge base of generic statements. *CoRR*, abs/2005.00660.

Anne Bosse. 2021. Generics: Some (non) specifics. *Synthese*, (5-6):14383–14401.

Greg N. Carlson, editor. 1977b. *Reference to Kinds in English*.

Greg N. Carlson and Francis Jeffry Pelletier, editors. 1995. *The Generic Book*. University of Chicago Press.

Gregory N. Carlson. 1977. A unified analysis of the english bare plural. *Linguistics and Philosophy*, 1:413–457.

Gustavo Cilleruelo, Emily Allaway, Barry Haddow, and Alexandra Birch. 2025. Generics are puzzling. can language models find the missing piece? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6571–6588, Abu Dhabi, UAE. Association for Computational Linguistics.

Ariel Cohen. 1999a. Generics, frequency adverbs, and probability. *Linguistics and Philosophy*, 22(3):221–253.

Ariel Cohen. 1999b. *Think Generic!: The Meaning and Use of Generic Sentences*. CSLI, Stanford.

Claudia Collacciani, Giulia Rambelli, and Marianna Bolognesi. 2024. Quantifying generalizations: Exploring the divide between human and llms' sensitivity to quantification. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11811–11822. Association for Computational Linguistics.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027.

Aurélie Herbelot and Eva Maria Vecchi. 2016. Many speakers, many worlds: Interannotator variations in the quantification of feature norms. *Linguistic Issues in Language Technology*, 13.

Daniel Kahneman. 2011. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, New York.

Matthew Kenney. 2023. arxiv_s2orc_parsed.

James Ravi Kirkpatrick. 2023. The dynamics of generics. *Journal of Semantics*, 40(4):523–548.

James Ravi Kirkpatrick. 2024. Are generics quantificational? *Synthese*, 204(17).

Anna Kurek-Przybilski and Adam. 2022. Generics as a paradigm: A corpus-based study of norwegian.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

Sarah-Jane Leslie. 2007a. *Generics, cognition, and comprehension*. Ph.d. dissertation, Princeton University. Order No. 3256578.

Sarah-Jane Leslie. 2008. Generics: Cognition and acquisition. *Philosophical Review*, 117(1).

Sarah-Jane Leslie, Sangeet Khemlani, and Sam Glucksberg. 2011. Do all ducks lay eggs? the generic overgeneralization effect. *Journal of Memory and Language*, 65(1):15–31.

Sarah-Jane Leslie. 2007b. Generics and the structure of the mind. *Philosophical Perspectives*, 21:375 – 403.

David Lewis. 1975. Adverbs of quantification. pages 5–20.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *CoRR*, abs/1510.03055.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

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

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark. *Preprint*, arXiv:2210.07316.

Eleonore Neufeld, Annie Bosse, Guillermo Del Pinal, and Rachel Sterken. 2025. Giving generic language another thought. *WIREs Cognitive Science*.

Anthony Nguyen. 2020. The radical account of bare plural generics. *Philosophical Studies*, 177(5):1303–1331.

Bernhard Nickel. 2016. *Between Logic and the World: An Integrated Theory of Generics*. Oxford University Press UK, Oxford, GB.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only. *Preprint*, arXiv:2306.01116.

Uwe Peters, Henrik Sherling, and Benjamin Chin-Yee. 2024. Hasty generalizations and generics in medical research: A systematic review. *PLOS ONE*, 19.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *CoRR*, abs/2003.07082.

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

Sello Ralethe and Jan Buys. 2022. Generic overgeneralization in pre-trained language models. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3187–3196, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

Luca Soldaini and Kyle Lo. 2023. peS2o (Pretraining Efficiently on S2ORC) Dataset. Technical report, Allen Institute for AI. ODC-By, `https://github.com/allenai/pes2o`.

Rachel Sterken. 2015. Generics in context. *Philosophers' Imprint*, 15:1–30.

Preston Stovall. 2019. Characterizing generics are material inference tickets: A proof-theoretic analysis. *Inquiry: An Interdisciplinary Journal of Philosophy*.

Michael Henry Tessler and Noah D. Goodman. 2016. The language of generalization. *CoRR*, abs/1608.02926.

Guy Tevet and Jonathan Berant. 2021. Evaluating the evaluation of diversity in natural language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 326–346, Online. Association for Computational Linguistics.

Yury Tokpanov, Beren Millidge, Paolo Glorioso, Jonathan Pilault, Adam Ibrahim, James Whittington, and Quentin Anthony. 2024. Zyda: A 1.3t dataset for open language modeling. *Preprint*, arXiv:2406.01981.

Thomas P. Urbach and Marta Kutas. 2010. Quantifiers more or less quantify on-line: Erp evidence for partial incremental interpretation. *Journal of Memory and Language*, 63(2):158–179.

A Training and evaluation of the generics classifier

Training. We build the generics classifier by training a first iteration on generics from GENERICSKB and then refining it iteratively. We make the training set more complete by adding examples the classifier struggles on from the candidate bare plurals, thus covering difficult and corner cases. We synthetically augment this challenging datapoints with the prompts in Appendix B. Table A.1 shows the final distribution of the training dataset, which trains a classifier that reaches 0.97 F-1 score in our 3622 sentences evaluation set.

Origin	Sentences
GENERICSKB (generics)	2500
Synthetic non-generics	2039
Non-generics from data	310
Generics from data	61

Table A.1: Composition of the ROBERTA classifier training data.

Evaluation data. We evaluate the generics classifiers in CONGEN for positive examples and a synthetic negative examples generated with GPT-4 (OpenAI et al., 2024). We include the quantified sentences in CONGEN by removing the quantifier (*most tigers hunt rabbits* becomes *tigers hunt rabbits*). The negative (non-generic) sentences are designed to be challenging for a generics classifier (details are available in Appendix B). The final test set includes 3622 test sentences: 2873 generics and 749 non-generics.

B Synthetic adversarial non-generic bare plurals generation

We combine variations of the following prompts to generate synthetic data based on difficult examples in the data, where iterations of the generics classifier struggle. We also focus on filtering out some examples undetectable to the synthetic filtering step, such as sentences with the title section present (for example, *Introduction Transformers are function approximators*). We use some of the synthetic examples generated for the training and some for the evaluation of the classifier.

Prompt#1.
```
Task: generation of declarative sentences indicative that are not generic. The sentences generated should not be generic sentences, even if they share features with them. The following examples are non-generic sentences, or sentences that do not begin with the generic sentence.

Examples:
{ list of examples}

Based on the previous examples, generate 100 non-generic sentences using a wide range of vocabulary and basing the generated sentences on the types of syntax in the examples, and other varied
```

syntactic constructions similar to bare plurals, such as adding elements that make it so that the generic sentence is not at the beginning or is not grammatical. The setences cannot begin with a generic, such as "tigers have stripes" or "nerves carry messages throughout the body", but rather existentials, ungrammatical or beginning with a section title. Generate the examples in the format of a python list of strings.

Prompt#2. Task: generate existential sentences that syntactically resemble bare plural generic sentences. For examples are sentences that talk about figures, equations, examples and studies in scientific articles, such as "Blue arrows indicate acceleration", "Examples of this are equations 2 and 4" or "Studies show this phenomena happens often". Can you generate 100 sentences like these in a python list sentence. Make them with varied lengths and lexically varied, and make sure they are clearly not generic, for example by referencing figure numbers etc.

Prompt#3. Generate 10 sentences that have a similar structure than the following example. Return the results in the format of a python list.

Example: Processes are made of repetitive...

C Sentence Length in MGEN

The 20,000 sampled documents sampled from MGEN yield a total of 4,202,451 sentences.

Dataset	Average	Median
MGEN (generics)	16.65 ± 8.2	15
MGEN (documents)	24.75 ± 29.3	21
GENERICSKB-BEST (natural)	9.66 ± 3.66	10

Table C.2: Average and median length across datasets.

D Annotation of MGEN

These are the instructions and examples annotators received:

- Assign the label “Generic”, “Particular” or “Unclear” to each sentence in your sheet.
- “Generic” sentences make a broad statement that applies to members of a category or group in general. For example, *Birds fly*, *German shepherds are loyal*, *Well-maintained public parks attract visitors all year-round*. Even if the group is very specific, such as *Red birds with long beaks that live in the jungle fly*, as long as it does not appear like the text refers to specific individuals in the context, label it as a generic.
- “Particular” sentences talk about a specific set of individuals or events. They usually provide information about one or a few individuals in a group: *This bird can fly*, *Dogs are in the front lawn*. These are sentences that talk about particular things in a context: *Units are in kilograms*, *Arrows indicate acceleration* would not be generics as they only make sense when refering to a specific table or plot. *German shepherds outside the house are loyal* is also not a generic, as it refers to specific german shepherds.
- In case of subsentences, focus only on the first subsentence: *Birds fly and this parrot speaks* would still count as generic even if "this parrot speaks" is not a generic since it refers to a particular parrot.
- Do not worry if you are unsure about whether a sentence is “Generic” or “Particular”. In this case, or if the sentence is grammatically incorrect, please use the “Unclear” label. Use also "Unclear" if you are not sure, you would need more context to answer or if the first words in the sentence are not a generic (for example: *In any case, birds fly*)
- For more examples, have a look at the annotated sentences in red. Thank you for your participation!

They also had the following examples:

- Tigers have stripes. *Generic*
- Tigers have stripes, they are cats and the ones we have here are violent. *Generic*
- Those tigers have stripes. *Particular*
- Tigers, which are part of the Felidae family, have stripes. *Generic*
- Tigers in this zoo are violent. *Particular*

- Tigers in zoos are violent. *Generic*
- Tigers are in the front lawn. *Particular*
- Tigers are also like this. *Generic*
- Tigers share that characteristic with lions. *Generic*

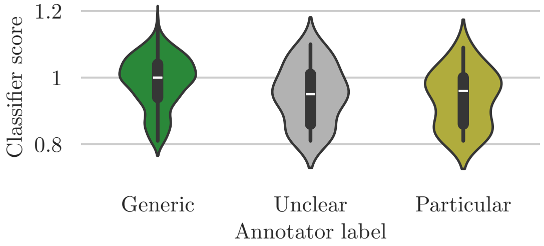


Figure D.1: Correspondence of human annotations with RoBERTa classifier scores.

E Composition of the MGen dataset

Table E.3 shows the millions of documents each component of Zyda has. Note that we only mine generics from about 23% of the dataset. The final amount of sentences in MGen by source is in Table E.4.

Finally, Table E.5 shows the top 50 common words for generics in MGen, naturally occurring sentences in GenericsKB-Best and 100,000 documents sampled from the contexts in MGen.

Source	Total Documents	Origin
RefinedWeb	$920.5M$	Internet
SlimPajama	$142.3M$	Internet
The Pile	$64.9M$	Varied
peS2o	$35.7M$	Academic
arXiv	$0.3M$	Academic

Table E.3: Information on the components of Zyda we run the generics pipeline on.

Source	Sentences
RefinedWeb	1,270,280
The Pile	1,019,687
SlimPajama	993,373
peS2o	796,334
arXiv	67,086

Table E.4: Combined statistics for MGen by source.

F Data samples

MGen (generics)		GenericsKB-Best		MGen (100k documents)	
Word	**Count**	**Word**	**Count**	**Word**	**Count**
people	200946	also	23933	also	942208
also	183012	water	20301	data	879361
cells	96700	one	18145	using	780702
used	96104	people	16598	one	767704
different	94097	many	12452	model	735504
use	92326	important	12417	used	727311
like	89778	life	11283	two	653421
one	84314	plants	10967	different	591577
make	74173	cause	10933	figure	587311
high	70107	common	10923	time	585129
many	70083	used	10715	study	584773
need	70010	body	10344	results	576442
women	68460	use	10074	may	568490
time	64141	different	10036	cells	539390
children	61270	food	9964	al.	535876
well	60362	animals	9315	however	477362
systems	60005	energy	8891	use	476105
tend	57323	human	8886	number	474336
important	56710	cells	8858	system	468788
provide	56523	form	8660	analysis	446709
work	55676	time	8478	first	445497
less	50941	children	7757	fig	438667
good	50521	women	7618	based	385968
much	48714	blood	7147	models	373924
get	47917	light	7109	high	372224
large	47588	small	7086	function	371581
small	47149	disease	6953	learning	370877
water	46181	world	6884	information	370467
way	45507	cancer	6653	case	356658
even	44487	natural	6583	set	351422
common	44330	like	6527	shown	349042
may	43538	part	6452	table	348287
patients	43443	often	6257	cell	341799
likely	43303	large	6220	new	334611
higher	43208	make	6199	given	330825
health	42758	high	6148	well	326821
help	41548	air	6017	studies	325837
men	40689	health	5982	patients	325434
system	40548	live	5889	research	321275
known	40036	two	5774	found	319645
play	39813	way	5503	could	317444
two	38604	well	5478	due	314760
human	38571	means	5464	see	312387
life	38428	occurs	5447	systems	306782
data	37663	process	5403	energy	304915
great	37612	soil	5397	thus	303428
form	37517	occur	5373	method	299352
new	37113	growth	5157	process	298258
n’t	36267	work	5145	group	290830
social	36212	system	5046	would	289965

Table E.5: Top 50 common words in generic sentences from MGen and GenericsKB-Best.

Bare plural	**Source**
Solid lines are the analytical results (Eqs.	arXiv
State police report 30 year old Kira Zink was headed south …	SlimPajama
Svp binding sites are underlined.	The Pile
COST: Entries start at $10; MORE INFO TUESDAY, DECEMBER 24…	SlimPajama
Online master's programs close on May 5th and August 19th.	SlimPajama
Tickets cost £12 (students £5, under 18s go free)…	RefinedWeb

Table F.6: Examples of existential (non-generic) bare plurals from ZYDA. Dots (…) indicate the example was truncated.

Sentences	**Source**
Horses are mammals	(Carlson, 1977)
Horses are larger than mules	(Carlson, 1977)
Elephants are easily trained	(Carlson, 1977)
Mosquitoes carry the West Nile virus	(Leslie, 2008)
Cats have whiskers	(Leslie, 2008)
Peacocks have fabulous blue tails	(Leslie, 2008)
Diamonds are valuable	(Nickel, 2016)
Elephants live in Africa or Asia	(Nickel, 2016)
Coke bottles have short necks	(Nickel, 2016)
Cabs are yellow	(Sterken, 2015)
Birds lay eggs, but mammals don't. Mammals give birth to live young.	(Sterken, 2015)
Lottery tickets are losers	(Sterken, 2015)

Table F.7: Some generics that serve as running examples in the literature.

Text	Label 1	Label 2	Score
Textbooks provide templates for proper procedure: the who, why, what, and where of the story.	Generic	Generic	0.91
Flatforms are comfy because of the uniform thickness of the heel and at the same time practical and easy to style in the morning with jeans and T-shirts and in the evening with Oversized Dresses.	Generic	Generic	0.90
Males have two sex organs, known as hemipenes, which are normally kept within the body, but are everted from his vent for mating.	Unclear	Generic	1.06
Cash crops are called commercial or commercial crops.	Generic	Generic	1.03
Oil-based primers are also very good remedies for covering staining on walls and ceilings that have oil-based paints.	Generic	Generic	1.02
Thin clients are less intelligent terminals that connect to applications hosted on a remote computer.	Unclear	Generic	1.03
Thicker greens such as romaine or bib lettuce are better for salads that will have a lot of meat or chunky vegetables.	Generic	Generic	1.07
JWs today have a similar command structure to promote uniformity rather than truth and love, in every element of a Christians life.	Generic	Generic	0.95
People realize that the best way to control their housing costs is ownership.	Generic	Generic	1.03
People who wish to argue against Spiritualism are quite sure, as a rule, that media will descend to any trickery and cheating for the sake of gain.	Generic	Generic	0.93
Red d'Anjou pears are excellent for fresh eating, poaching, cooking and all types of baking.	Generic	Generic	0.95
Powerful computing systems also require high speed access to large data storage systems.	Generic	Generic	0.95
Filipinos of Hispanic ancestry form a minority in the Philippine population.	Generic	Generic	1.06
IMTs operate in various ways.	Generic	Unclear	0.99
Weak institutions lead to weak coordination and fragmented interventions that often prove ineffective.	Generic	Generic	1.04
Ventilation flaps are used in the air ducts of heating and ventilation systems or air conditioning systems in an automobile and are usually adjusted via Bowden pull mechanisms or mechanical transmissions.	Generic	Generic	1.05
Quantum computers promise to directly simulate systems governed by quantum principles, such as molecules or materials, since the quantum bits themselves are quantum objects.	Generic	Generic	1.04
Pair bonds are monogamous and seasonal. 3–6 eggs are incubated by the female only, but the chicks are usually brooded and fed by both birds.	Generic	Generic	1.03
Puppets are fun to include too.	Particular	Unclear	0.86
Parenchyma cells are also responsible for healing in the plant - this tissue can go through cell division and regenerate when needed.	Generic	Generic	1.03
Conventional linear synchronous motors have issues of high manufacturing cost of the stator and high magnetic loss.	Generic	Generic	0.99
Traditions are a vital a part of the Italian culture and naturally, weddings have their very own.	Generic	Unclear	0.92
Calm dog breeds include Great Danes, Great Pyrenees, Basset Hounds, Shih Tzus, and Pugs.	Unclear	Unclear	0.84
First thoughts are proverbially the best; at all events, they are the bravest.	Unclear	Generic	0.96
Bursts are by definition variable, as temperature evolution due to thermonuclear burning and then cooling drives the fast increase and then slower decrease in X-ray flux.	Particular	Generic	0.97
People are under pressure to make the systems efficient, but they are expected to keep the system safe, which inevitably introduces inefficiencies.	Particular	Generic	0.91
Police officers are human beings, and many of them understand that the pressures of everyday life can sometimes lead good drivers to make bad decisions.	Generic	Generic	1.11
Self-induction habits are oft described as a compulsive behavior, with magnetic-like attraction to light sources commonly reported [9].	Generic	Generic	0.88
Gastroenterologists, infectious disease specialists, hepatologists, and even some nurse practitioners commonly manage cases of Hep C.	Unclear	Generic	1.1
Natural degradable polymers and their composites are amongst these materials.	Particular	Generic	0.84
Involving surrounding tissue structures, tonsillar tumours often infiltrate the soft palate, the base of the tongue, the lateral pharyngeal wall and medially the parapharyngeal space as well as the vascular sheath.	Generic	Unclear	0.83
Caries are understood to result from the accumulation of plaque on the teeth and the production of organic acids (plaque acids) when plaque microorganisms ferment sugars and starches in food.	Generic	Generic	1.06
Female beetles deposit their eggs singly on the legume seeds.	Generic	Generic	1.06

Table F.8: 33 examples from MGEN generics with both annotations and scores.

Length	Generic	Source	Score
3	Words have power.	RefinedWeb	0.98
4	Democrats are control freaks.	The Pile	1.01
5	Children learn what they live.	The Pile	1.08
6	Ghosts represent a post-death human consciousness.	SlimPajama	1.02
7	Color and pictures are fun and vibrant.	RefinedWeb	0.82
8	More complex bytecodes trap to a software routine.	peS2o	0.85
9	Males tend to be more affected by the disease.	SlimPajama	0.99
10	Triggers cause individuals to become ineffective and produce negative energy.	The Pile	1.02
11	Professional massage therapists relieve tired muscles and alleviate pain in customers.	RefinedWeb	0.97
12	American workers produce sophisticated goods or investment opportunities at lower opportunity costs.	SlimPajama	1.06
13	Insurance companies reward property owners who personal their house totally free and obvious.	RefinedWeb	1.0
14	Alkaline phosphatases carry out hydrolase/transferase reactions on phosphate-containing substrates at a high pH optimum.	The Pile	1.0
15	Stimulants are substances that raise the levels of physiological or nervous activity in the body.	RefinedWen	1.04
16	Areas along large rivers are commonly inhabited by baldcypress, water tupelo, water elm, and bitter pecan.	The Pile	0.94
17	Sports fans are far more familiar with NBC Sports, which televises everything from Super Bowls to Olympics.	The Pile	0.96
18	Keto dieters love exogenous ketones because they help fight the keto flu and get you quickly into ketosis.	The Pile	1.07
19	Insects evolve adaptations allowing them to eat specific species of plants, while being unable to eat most other plants.	RefinedWeb	1.04
20	Extractive methods, such as lipoplasty (liposuction) or local excision, are methods whereby fat is mechanically removed from areas of interest.	The Pile	0.96
21	Factory-terminated systems are also the only viable solution to the extremely low-loss systems that are required to support high-speed optic links.	RefinedWeb	0.86
22	Small Business consultants typically develop relationships with their customers and often correspond by e-mail with their customers and return customers' phone calls.	The Pile	0.99
23	Initial parton showers interact with the medium via collisional and radiative processes that cause dissipation and redistribution of energy inside the parton shower.	peS2o	0.93
24	Green superfoods have the highest concentrations of simply digestible nutrients, fat burning compounds, nutritional vitamins and minerals to safeguard and mend your body. !	RefinedWeb	0.87
25	Punitive damages are awarded to punish a defendant for particularly egregious conduct, and to serve as a deterrent to future conduct of the same type.	The Pile	0.96

Table F.9: Examples of generics from MGEN at different sentence lengths.

Intonation as a quantifier-free logical interpretation of metrical and prosodic structure

Hyunjung Joo and Adam Jardine
Department of Linguistics
Rutgers University
{hyunjung.joo,adam.jardine}@rutgers.edu

Abstract

This study views *intonation* as a quantifier-free (QF) logical interpretation of a metrical and prosodic structure. Under logical transductions, tones in intonational melodies can be interpreted as literal copies of prosodic elements, with their association to TBUs being a local process. The head-prominence intonational pattern in American English can be defined by copying both accented syllables (heads) and phrasal boundaries, whereas the edge-prominence pattern in Seoul Korean was defined by copying only phrasal boundaries (edges). For Tokyo Japanese, lexical pitch accents are defined by copying accented moras, and post-lexical tones by copying phrasal boundaries. This QF interpretation of intonation structure enabled restrictive predictions about computational complexity and typology of intonation.

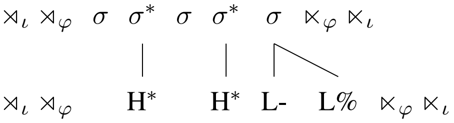
1 Introduction

How can we define what it means to be a possible intonational pattern in a precise way? Here we view *intonation* as a quantifier-free (QF) logical interpretation of a metrical and prosodic structure (Chandlee and Lindell, to appear; Strother-Garcia, 2019). Importantly, in this framework, tones in intonational melodies are viewed as literal copies of elements in the metrical and prosodic structure, such as accented syllables or phrasal boundaries, and they are always linked *locally* to their tone-bearing units (TBUs). Importantly, because QF is a very weak logic, a theory of intonation built around QF interpretations makes strong predictions about what is a possible intonational pattern. We show support for these predictions by showing that major intonational patterns are QF interpretations.

In the Autosegmental-metrical (AM) theory of intonation (e.g., Pierrehumbert, 1980), intonation can be defined as a sequence of Highs (Hs) and Lows (Ls). The tones in intonation are associated with their TBUs within the nested prosodic domains. Languages may vary depending on which prosodic elements, such as prominent syllables and/or phrasal boundaries, are used for intonation.

For example, in American English, intonational tones are associated with *metrically strong positions* and phrasal boundaries in an utterance. (1) shows an utterance ”an orange ball gown” produced with intonation. Within an intermediate phrase (ip; $\rtimes_\varphi/\ltimes_\varphi$), pitch accents (H*) are associated with accented syllables (σ^*) and a phrasal tone (L-) is associated with the final syllable of the ip. Then, within an Intonational Phrase (IP; $\rtimes_\iota/\ltimes_\iota$), a boundary tone (L%) is associated with the final syllable of the IP.

(1) [[ən ɔɹɪnʤ bɔl gaʊn]$_\varphi$]$_\iota$


$\rtimes_\iota$ $\rtimes_\varphi$ σ σ^* σ σ^* σ $\ltimes_\varphi$ $\ltimes_\iota$
$\rtimes_\iota$ $\rtimes_\varphi$ H* H* L- L% $\ltimes_\varphi$ $\ltimes_\iota$

Jardine (2017) showed that autosegmental representation of lexical tones and their TBUs is an *interpretation* of the toned syllables in the input structure, using *logical transductions* (Courcelle, 1994; Engelfriet and Hoogeboom, 2001; Filiot and Reynier, 2016). Also, the tone–TBU association patterns in tonal languages have been studied in terms of their local nature and computational complexity (Chandlee and Jardine, 2019a; Chandlee and Jardine, 2021; Koser et al., 2019). Then, how can we define the autosegmental representation of intonation using logical interpretation and what does this say about the computational nature of intonation?

We extend Jardine (2017) and Strother-Garcia (2019) by viewing AM representations as additional structure imposed on an input string. In doing so, we find that intonational tones and their associations with TBUs are always local to accents and boundaries if we make reference to a metrical grid and a prosodic structure. That is, the accented

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 261-270.
Eugene, Oregon, July 18-20, 2025

syllables and boundaries in the input structure can be interpreted as intonational tones in the output structure, which are always linked to their TBUs that are near the accents and boundaries.

Also, there exists another evidence supporting the QF logical interpretation of intonation. Not every logically possible intonational pattern is attested. For example, there are no patterns like Midpoint Pathology (Hyde, 2008; Eisner, 1997), in which tones are associated to a center-most TBU, for the intonational patterns. Computing such a tonal sequence demands memory proportional to the sequence length, exceeding the regular complexity bound of phonology (Heinz and Idsardi, 2011; Johnson, 1972; Kaplan and Kay, 1994) and thus far exceeding the power of QF.

Therefore, we can start with a hypothesis that intonation can be a QF logical interpretation of a metrical and prosodic structure, by examining three different intonation patterns: a *head-prominence* language, American English; an *edge-prominence* language, Seoul Korean; a *lexical pitch accent* language, Tokyo Japanese.

Based on this local nature of intonation, we can posit a theory that makes restrictive predictions about the intonational typology and measure the complexity of intonational structures, as the connections between logical interpretations and computational complexity are well-studied (Filiot and Reynier, 2016). This has been fruitfully applied to the study of phonological representations (Strother-Garcia, 2019; Jardine, 2017; Jardine et al., 2021).

2 Preliminaries

2.1 String models and logic

We define a finite alphabet of symbols as Σ and the set of all strings over σ^*. We use two boundary symbols $\rtimes, \ltimes$ to indicate the beginning and the end of strings. For example, for $\Sigma = \{C, V\}$, $\rtimes CCV \ltimes$ is a string over Σ delineated with boundaries.

We can describe strings and other structures with *models* in the following way (Enderton, 2001; Libkin, 2004). A *signature* is a set $\{R_1, ..., R_m, f_1, ..., f_n\}$ of named relation and function symbols. (We do not use signatures with constant symbols.) A *model* is thus an instantiation $\langle D; R_1, ..., R_m, f_1, ..., f_n\rangle$ of this set of relations and functions with a domain D of elements.

For example, in strings over an alphabet Σ, we can describe them with a signature $\{P_\sigma \in \Sigma, p, s\}$. where each $P_{\sigma\in\Sigma}$ is a unary relation that refers to a set of positions over the domain D for each σ in the alphabet. The predecessor and successor functions are p and s that return the immediately preceding and immediately following element in the string, respectively. For example, in $\{P_C, P_V, p, s\}$, P_C and P_V refer to the sets of positions over the domain D for C and V, respectively. With this signature the string $\rtimes CCV \ltimes$ can be defined with the following string model:

$$\begin{aligned}
\langle D = \{0,1,2,3,4\};\\
P_C = \{1,2\}, P_V = \{3\}, P_\rtimes = \{0\}, P_\ltimes = \{4\};\\
p = \{(0,1),(1,2),(2,3),(3,4)\};\\
s = \{(1,0),(2,1),(3,2),(4,3)\}\rangle
\end{aligned}$$

From a signature we immediately get a *first order* (FO) predicate logic in the usual way. Briefly, x, y, ... denoting *variables* that range over positions in a string; $\sigma(x)$ for each $\sigma \in \Sigma$ denoting *atomic predicates* which are true when x is interpreted as positions in the unary relation P_σ of a model; and FO formulae are are built recursively out of the logical connectives $\neg, \vee, \wedge, \rightarrow$ and quantifiers $\exists, \forall$. A *free variable* is a variable not bound by a quantifier. QF is the fragment of FO in which no quantifiers appear.

2.2 Logical transductions

Based on the input string that we've just defined, we can build a larger model using *logical transductions* (Courcelle, 1994; Engelfriet and Hoogeboom, 2001; Filiot and Reynier, 2016). We *interpret* the input structure into a finite number of *copies* in the output structure, using FO formulas. Via a logical transduction τ, the domain of the input structure (Σ) in the signature ($\mathcal{S}_i$) is extended in the output structure (Γ) in the signature ($\mathcal{S}_o$), which is represented with copies (Cs) of the input domain. Following Strother-Garcia (2019), we use syllable structure as an example, as shown in Figure 1.

The output structure Γ is defined with relations R' satisfied for any transduction τ if $\langle D'$; R'_1, ..., $R'_n\rangle$ is based on the input signature $\mathcal{S}_i$. For instance, $C_o(x) \overset{\text{def}}{=} C_i(x)$ means a consonant x appears in the output if and only if it exists in the input.

The domain D' of the output structure Γ is expanded by copying input elements n times, creating n copies of each input element. Unary relations R' are represented as R'^n for $n \in C$ (e.g., $C_o^0(x)$ denotes a consonant in the 0th copy). Binary relations R' are represented as $R'^{m,n}$ for $m, n \in C$

(e.g., $\mathcal{A}_o^{0,1}(x, y)$ denotes an association between x in the 0th copy and y in the 1st copy). The order p_o of the output structure Γ over the domain D' is defined separately for the 0th copy and the 1st to nth copies ($n > 0$), while preserving the order of the input structure Σ over D for both copies, following Chandlee and Jardine (2019b).

For the 0th copy, $p_o(x^0) \overset{\text{def}}{=} \mathrm{p}_i(x)$, such that the output order p of the elements in the 0th copy of Γ works the same as that in the input structure Σ, just like an identity function. For all the set of nth copies except for the 0th copy, $p_o(d^n) \overset{\text{def}}{=} \mathrm{e}^m (n, m > 0)$ if and only if $(p(d) = e) \vee (d \approx e \wedge p(n) = m)$. That is, for any elements $d, e \in D$ and for the copies $m, n \in C$, the element e^m precedes the element d^n in the output, with two conditions. The first condition is that if there are two distinct domain elements, we follow the order of the elements, such that if the element e precedes the element d in the input, the element e^m always precedes the element d^n in the output. However, the second condition is that if there are two identical domain elements in different copies, we follow the order of the copies such that if the mth copy precedes nth copy, the element e in the mth copy always precedes the element d in the nth copy.

From the $\rtimes CCV \ltimes$ string, we can build a syllable structure in the output, using logical transductions. The input strings are copied twice in the output (C0, C1) and each node with a free FO variable x is defined accordingly. The order of copies in the input, as determined by the predecessor and successor functions p_i and s_i, is preserved in the output using p_o and s_o.

$$C_o^0(x) = C_i(x) \quad V_o^0(x) = V_i(x)$$
$$\rtimes_o^0(x) = \rtimes_i(x) \quad \ltimes_o^0(x) = \ltimes_i(x)$$
$$\sigma_o^1(x) = V_i(x)$$
$$\begin{aligned}\mathcal{A}_o^{0,1}(x, y) = {} & C_i(x) \wedge V_i(y) \wedge y \approx s(s(x))) \vee \\ & (C_i(x) \wedge V_i(y) \wedge y \approx s(x)) \vee \\ & (V_i(x) \wedge V_i(y) \wedge y \approx x)\end{aligned}$$

In the first copy (C0), every C and every V in the input has one copy with the same label in the output. Also, boundaries in the output, $\rtimes_o^0(x)$ and $\ltimes_o^0(x)$ are the same as in the input. Importantly, for the second copy (C1), syllables in the output, $\sigma_o^1(x)$, is defined from a vowel in the input, $V_i(x)$, showing that every syllable is a *reflection* of nucleus.

Then, we can establish some relations between the output copies to build phonological structures.

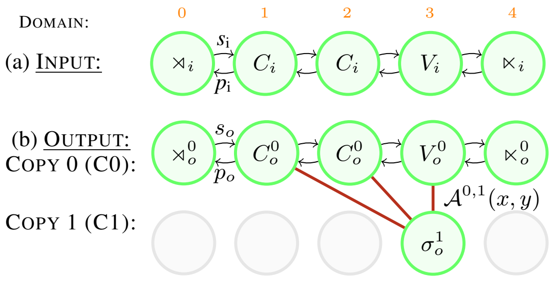


Figure 1: The illustration of a logical transduction from the input string $\rtimes CCV \ltimes$ to the output syllable structure.

$\mathcal{A}_o^{a,b}(x, y)$ defines an association relationship between the output copies over two free variables x and y, where a and b indicate the copies in the output. $\mathcal{A}_o^{0,1}(x, y)$ associates the two Cs and V in C0 with the syllable in C1, respectively. In this way, phonological structure building can be seen as an interpretation of a more basic structure.

Defining phonological processes with logical transductions allows us to measure computational complexity within the regular upper bound of phonology. Chandlee (2014) and Chandlee and Heinz (2018) showed that local phonological processes can be defined using input strictly local (ISL) functions, which are a proper subset of regular functions and are characterized by quantifier-free (QF) first-order logic. Chandlee and Jardine (2019b) showed that the subsequential functions for both local and long-distance phonological processes can be better characterized using QF first-order logic with a least fixed-point operator (QFLFP), further restricting them to a subset of the subsequential functions. As most phonological mappings are ISL (Chandlee, 2014; Chandlee and Heinz, 2018) and thus QF-definable (Chandlee and Lindell, to appear), a strong initial hypothesis for tone-TBU mappings in intonation is that they should be QF-definable. We investigate this hypothesis below.

3 Intonation as quantifier-free interpretation

Now turning to the intonational structures, we define a logical interpretation for intonation. Importantly, tones in intonational melodies are viewed as *copies* of elements in the metrical and prosodic structure, such as accented TBUs or boundaries. The source of intonational melodies is computationally defined as prosodic elements, but they are associated with their *local* TBUs in order to be realized as the actual tones.

Intonation involves two key stages of transduction: first, creating tonal slots with unspecified tones (Ts) via a *melodic transduction*, and second, filling these slots with specified tonal sequence with Hs and Ls via a *declarative meaning transduction*. While this section primarily focuses on outlining the properties and relations for melodic transduction, the details of the meaning transduction will be specified for each intonational pattern following the melodic stage.

For the melodic transduction, the input signature ($\mathcal{S}_i$) is $\{\sigma, \sigma^*, \ltimes_\varphi, \ltimes_\iota, \ltimes_\varphi, \ltimes_\iota, p, s, p^*, s^*\}$ and the output signature ($\mathcal{S}_o$) is $\{\sigma, \sigma^*, T, T^*, \ltimes_\varphi, \ltimes_\iota, \rtimes_\varphi, \rtimes_\iota, \mathcal{A}, p, s, p^*, s^*\}$, where each property and relation symbol in the signature is as follows: σ and σ^* for TBUs; $\rtimes_\varphi$ and $\ltimes_\varphi$ for ip boundary; $\rtimes_\iota$ and $\ltimes_\iota$ for IP boundary; T for tones other than pitch accent tones (nonstarred tones); T^* for pitch accent tones (starred tones). $\mathcal{A}$ is a binary association relation for tone and TBU.

For the unary relations, we can find the set of positions for each symbol with a variable x in the input structure. For example, $\sigma(x)$ is true when x is a syllable; $T(x)$ is true when x is a tone, etc.

As for the binary relations, in addition to p and s, we also define special *predecessor* and *successor* functions, p^* and s^*, to define the relations in the tier that is projected from the set of the selected elements such as metrically strong TBUs and phrasal boundaries. We use two tiers to represent a metrical grid: one for all the strings and the other for the starred elements and phrasal boundaries, as shown in Table 1. While the *nonstarred* function $p(x)$ works locally on the first tier, the *starred* function $p^*(x)$ works locally in the second tier. Similarly, $s(x)$ and $s^*(x)$ work the same way but in different directions.

$p^*(x)$	$\rtimes_\iota$	$\rtimes_\varphi$		σ_*		σ_*		$\ltimes_\varphi$	$\ltimes_\iota$
$p(x)$	$\rtimes_\iota$	$\rtimes_\varphi$	σ	σ	σ	σ	σ	$\ltimes_\varphi$	$\ltimes_\iota$
	$\rtimes_\iota$	$\rtimes_\varphi$	ən	ɔ*	ɹɪnʤ	bɔl*	gaʊn	$\ltimes_\varphi$	$\ltimes_\iota$

Table 1: A metrical grid using a tier-based representation.

Now, we will now look at three case studies, each focusing on a different intonational pattern.

3.1 American English

3.1.1 Basic intonational pattern

American English is a *head-prominence* intonational language (Beckman and Pierrehumbert, 1986), where *metrically strong positions* receive pitch accents in a phrase. For example, as shown in (2), the accented syllables (σ^*) are associated with pitch accents (H*) within an ip. A phrase tone (L-) is also associated at the right edge of the ip. Within an IP, the largest prosodic domain, a boundary tone (L%) is also associated with the right edge of the IP. The actual f0 contour of an English declarative for (2) is provided in Figure 2.

(2) $\rtimes_\iota \quad \rtimes_\varphi \quad \sigma \quad \sigma^* \quad \sigma \quad \sigma^* \quad \sigma \quad \ltimes_\varphi \quad \ltimes_\iota$

H* H* L- L%

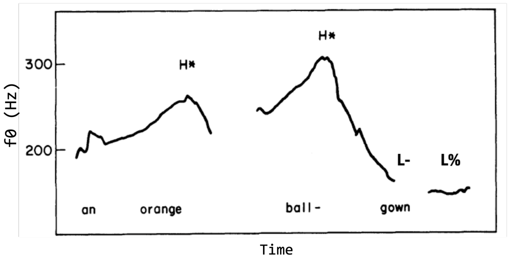


Figure 2: An actual f0 contour of a declarative intonational pattern in American English, extracted from Beckman and Pierrehumbert (1986).

3.1.2 Melodic transduction

Step 1: Copying The input is a string that consists of syllables (σ, σ^*) and boundaries ($\rtimes_\iota/\ltimes_\iota$, $\rtimes_\varphi/\ltimes_\varphi$). As defined in the formulas below, the outputs are four *copies* of the input, which are also illustrated in Figure 3. For the first copy (C0), everything in the input is copied such that syllables and ip and IP boundaries in the output are interpreted the same as those in the input.

$$\sigma_o^0(x) \overset{\text{def}}{=} \sigma_i(x) \qquad \sigma_o^{*0}(x) \overset{\text{def}}{=} \sigma_i^*(x)$$
$$\rtimes_{\varphi_o}^0(x) \overset{\text{def}}{=} \rtimes_{\varphi_i}^0(x) \qquad \ltimes_{\varphi_o}^0(x) \overset{\text{def}}{=} \ltimes_{\varphi_i}^0(x)$$
$$\rtimes_{\iota_o}^0(x) \overset{\text{def}}{=} \rtimes_{\iota_i}^0(x) \qquad \ltimes_{\iota_o}^0(x) \overset{\text{def}}{=} \ltimes_{\iota_o}^0(x)$$

In the formulas for the remaining copies (C1-C3) below, only starred syllables and boundaries are copied and interpreted as tones, reflecting the head-prominence characteristics of American English intonational patterns. In C1, starred syllables in the input, $\sigma_i^*(x)$, are realized as pitch accents in the output, $T_o^{*1}(x)$. In C2, ip boundary at the right edge, $\ltimes_{\varphi_i}(x)$, is realized as a phrasal tone, $T_o^2(x)$. In C3, IP boundaries at the left or right edge, $\ltimes_{\iota_i}(x) \vee \rtimes_{\iota_i}(x)$, are realized as boundary

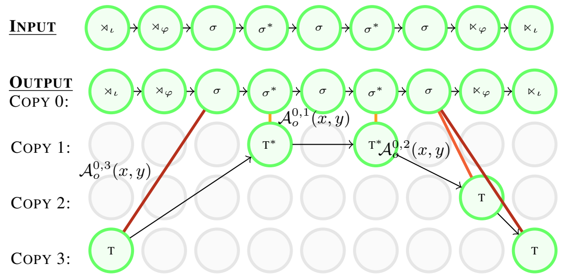

INPUT
OUTPUT
COPY 0:
COPY 1:
COPY 2:
COPY 3:
$\rtimes_\iota$ $\rtimes_\varphi$ σ σ^* σ σ^* σ $\ltimes_\varphi$ $\ltimes_\iota$
$\mathcal{A}_o^{0,1}(x,y)$
$\mathcal{A}_o^{0,2}(x,y)$
$\mathcal{A}_o^{0,3}(x,y)$
T*
T*
T
T
T

Figure 3: Melodic transduction of American English intonation.

tones, $T_o^3(x)$.

$$T_o^{*1}(x) \overset{\text{def}}{=} \sigma_i^*(x)$$
$$T_o^2(x) \overset{\text{def}}{=} \ltimes_{\varphi_i}(x)$$
$$T_o^3(x) \overset{\text{def}}{=} \ltimes_{\iota i}(x) \vee \rtimes_{\iota_i}(x)$$

Thus, tones in American English are direct copies of starred syllables and phrasal boundaries.

Step 2: Tone-TBU association Importantly, tones in the melodic tiers (C1-C3) are associated with syllables in the segmental tier (C0), as defined below. $\mathcal{A}_o^{0,1}(x,y)$ specifies the association between pitch accents in C1 and their TBUs in C0 if they are at the same position in the input. For phrasal and boundary tones, tones are linked to syllables near boundaries. Specifically, $\mathcal{A}_o^{0,2}(x,y)$ defines the association between phrasal tones at the right edge and the phrase-final syllables just before that edge. Similarly, $\mathcal{A}_o^{0,3}(x,y)$ links boundary tones to their TBUs: tones from the left edge are linked to the first syllable, while those from the right edge are linked to the last syllable in an utterance. Thus, tone-TBU association is computed using only predecessor or successor functions, showing a local logical characterization without quantifiers.

$$\mathcal{A}_o^{0,1}(x,y) \overset{\text{def}}{=} x \approx y$$
$$\mathcal{A}_o^{0,2}(x,y) \overset{\text{def}}{=} \sigma_i(x) \wedge \ltimes_{\varphi_i}(y) \wedge y \approx s(x)$$
$$\begin{aligned}\mathcal{A}_o^{0,3}(x,y) \overset{\text{def}}{=} &(\sigma_i(x) \wedge \rtimes_{\iota_i}(y) \wedge y \approx p(p(x)))\\ &\vee (\sigma_i(x) \wedge \ltimes_{\iota_i}(y) \wedge y \approx s(s(x)))\end{aligned}$$

3.1.3 Declarative meaning transduction

In the melodic transduction, we have made the slots for the tones that are associated with their TBUs. The remaining step is to compute the *meaning* of a declarative sentence in English, which is specified as H* H* L- L% tonal sequence in Figure 2. As shown Figure 4, we use another simple transduction that changes the unspecified tones (T/T^*) into actual tones (H^*/L), using these simple formulas: $H_o^*(x) = T_i^*(x)$ and $L_o(x) = T_i(x)$.

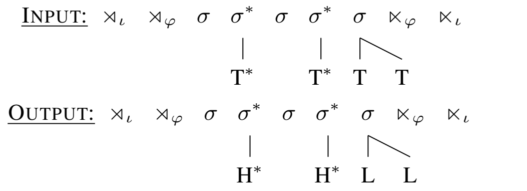

INPUT:
$\rtimes_\iota$ $\rtimes_\varphi$ σ σ^* σ σ^* σ $\ltimes_\varphi$ $\ltimes_\iota$
T* T* T T
OUTPUT:
$\rtimes_\iota$ $\rtimes_\varphi$ σ σ^* σ σ^* σ $\ltimes_\varphi$ $\ltimes_\iota$
H* H* L L

Figure 4: Declarative meaning transduction of American English intonation.

With these melodic and declarative transductions, we can logically define the intonational tones associated with their TBUs in the output based on the strings in the input.

3.1.4 Summary

Results showed that American English intonation can be defined as a QF logical interpretation of a metrical and prosodic structure. The melodies in the output were copies of starred syllables and boundaries in the input. Crucially, copying the starred syllables was able to capture the head-prominence characteristic in American English intonation, showing that the pitch accents in the melodies were the direct reflections of the *heads* of the prosodic unit – starred syllables. Also, the tone-TBU associations were defined *locally* from the input structure without using any quantifiers.

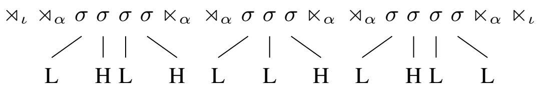
3.2 Seoul Korean

3.2.1 Basic intonational pattern

Seoul Korean is an *edge-prominence* intonational language (Jun, 2006), where phrasal boundaries are marked with prominence without any pitch accents. Basically, a typical tonal pattern is LH...LH in an Accentual Phrase (AP). But when the initial segment of an AP is an aspirated or a tense consonant, the tonal pattern is HH...LH. An Intonational Phrase (IP) consists of more than one AP.

In (3), LH tones are associated with the first two and last two syllables. However, in the final AP, the L% boundary tone overrides the phrase-final H tone at the end of an utterance. If a phrase has fewer than four syllables, one of the tones may not be realized. Edge tones—LH at the left edge and LH at the right edge—plays a crucial role in the intonational pattern of Seoul Korean. An actual f0 contour of a Korean declarative for (3) is provided in Figure 5.

(3)


$\rtimes_\iota$ $\rtimes_\alpha$ σ σ σ σ $\ltimes_\alpha$ $\rtimes_\alpha$ σ σ σ $\ltimes_\alpha$ $\rtimes_\alpha$ σ σ σ σ $\ltimes_\alpha$ $\ltimes_\iota$
L H L H L L H L H L L

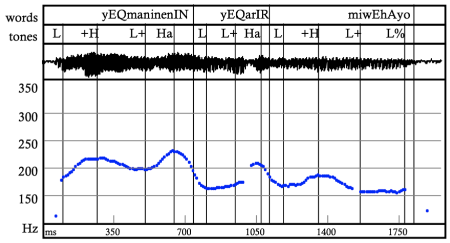


Figure 5: An actual f0 contour of one of the declarative intonational patterns in Seoul Korean, extracted from Jun (2006).

3.2.2 Melodic transduction

Step 1: Copying The input is a string that consists of syllables and boundaries. The outputs are four *copies* of the input, defined in the formula below. As shown in Figure 6, for C0, everything in the input is copied such that syllables and boundaries in the output is interpreted the same as those in the input. In Seoul Korean, the stiffness feature for aspirated or tense consonants ([+stiff]) is specified in the syllable, allowing retrieval during tonal contour computation (e.g., HH...LH).

$$\sigma_o^0(x) = \sigma_i(x) \qquad \sigma_{F_o}^0(x) = \sigma_{F_i}(x) \ (F = [+\text{stiff}])$$
$$\ltimes_{\alpha_o}^0(x) = \ltimes_{\alpha_i}^0(x) \qquad \ltimes_{\iota_o}^0(x) = \ltimes_{\iota_i}^0(x)$$
$$\rtimes_{\alpha_o}^0(x) = \rtimes_{\alpha_i}^0(x) \qquad \rtimes_{\iota_o}^0(x) = \rtimes_{\iota_i}^0(x)$$

As for C1-C3, only boundaries are copied and interpreted as tones, showing a crucial characteristic for the edge-prominence intonational property. Both C1 and C2 shows that AP boundaries at the left or right edge in the input, $\rtimes_{\alpha_i}(x) \vee \ltimes_{\alpha_i}(x)$, are realized as tones in the output, $T_o^1(x)$ and $T_o^2(x)$. A boundary at the end of an utterance, $\ltimes_{\iota_i}(x)$, is realized as a boundary tone, $T_o^3(x)$.

$$T_o^1(x) = \rtimes_{\alpha_i}(x) \vee \ltimes_{\alpha_i}(x)$$
$$T_o^2(x) = \rtimes_{\alpha_i}(x) \vee \ltimes_{\alpha_i}(x)$$
$$T_o^3(x) = \ltimes_{\iota_i}(x)$$

Thus, tones in Seoul Korean are simply direct copies of elements in the prosodic structure, which are only phrasal boundaries.

Step 2: Tone-TBU association The tones in the melodic tiers (C1-C3) are linked to the syllables in the segmental tier (C0). First, $\mathcal{A}_o^{0,1}(x,y)$associates a phrasal tone in C1 with either the first syllable of an AP or the second-to-last syllable of an AP in C0. $\mathcal{A}_o^{0,2}(x,y)$ links a phrasal tone in C2 to the second syllable of an AP, if it is preceded by a left edge of an AP or followed by the last syllable of an AP in C0. Finally, $\mathcal{A}_o^{0,3}(x,y)$ links a boundary tone in C3 to the last syllable before the boundary. The boundary tone in C3 overrides the AP-final phrasal tone in C2, reflecting the hierarchy of boundary tones over phrasal tones.

$$\begin{aligned}
\mathcal{A}_o^{0,1}(x,y) &= \sigma(x) \wedge (\rtimes_\alpha(y) \wedge \ y \approx p(x)) \\
&\quad \vee (\ltimes_\alpha(y) \wedge \ y \approx s(s(x))) \\
\mathcal{A}_o^{0,2}(x,y) &= \sigma(x) \wedge \neg(\ltimes_\iota(y) \wedge y \approx s(s(x))) \\
&\quad \wedge (\rtimes_\alpha(y) \wedge y \approx p(p(x))) \\
&\quad \vee (\rtimes_\alpha(y) \wedge y \approx s(s(x))) \\
\mathcal{A}_o^{0,3}(x,y) &= \sigma(x) \wedge \ltimes_\iota(y) \wedge y \approx s(s(x))
\end{aligned}$$

3.2.3 Declarative meaning transduction

After the melodic transduction, the unspecified tones (Ts) are filled with Hs and Ls for the declarative in Seoul Korean, as shown in Figure 7. The input signatures are $\{\sigma, \ltimes_\varphi, \ltimes_\iota, \rtimes_\varphi, \rtimes_\iota, T\}$ and the output signatures are $\{\sigma, \ltimes_\varphi, \ltimes_\iota, \rtimes_\varphi, \rtimes_\iota, H, L\}$. The formulas are as follows: $L_o(x) = T_i(x) \wedge (\rtimes_\alpha(p(x)) \vee \ltimes_\alpha(s(s(x))))$ and $H_o(x) = T_i(x) \wedge (\rtimes_\alpha(p(p(x))) \wedge \neg H(s(x))) \vee \ltimes_\alpha(s(x))$.

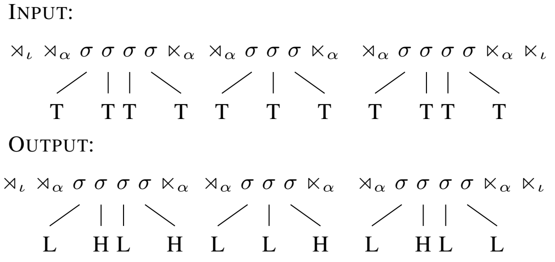

INPUT:
$\rtimes_\iota$ $\rtimes_\alpha$ σ σ σ σ $\ltimes_\alpha$ $\rtimes_\alpha$ σ σ σ $\ltimes_\alpha$ $\rtimes_\alpha$ σ σ σ σ $\ltimes_\alpha$ $\ltimes_\iota$
T T T T T T T T T T T
OUTPUT:
$\rtimes_\iota$ $\rtimes_\alpha$ σ σ σ σ $\ltimes_\alpha$ $\rtimes_\alpha$ σ σ σ $\ltimes_\alpha$ $\rtimes_\alpha$ σ σ σ σ $\ltimes_\alpha$ $\ltimes_\iota$
L H L H L L H L H L L

Figure 7: Declarative meaning transduction of Seoul Korean intonation.

3.2.4 Summary

Seoul Korean intonational pattern can be defined using logical interpretation of a prosodic structure. The melodies in the output were copies of *only* boundaries from the input, capturing the edge-prominence characteristic of Seoul Korean intonation. This reflects the edge tones as direct representations of phrasal edges. Similar to American English, the tone-TBU associations were defined *locally* from the input without quantifiers.

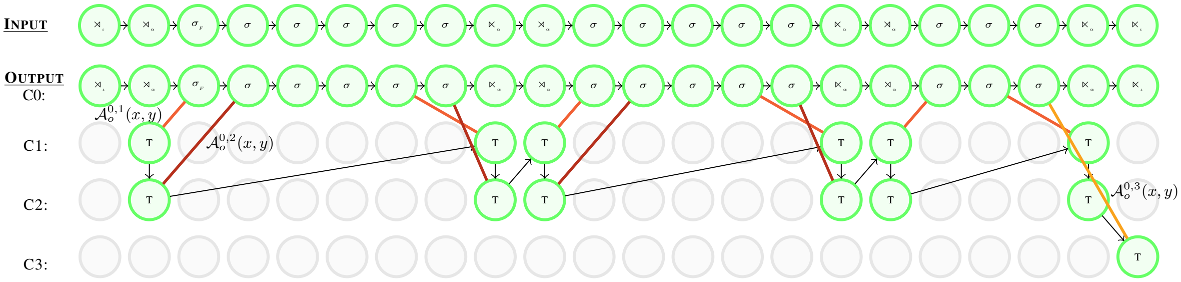

INPUT
OUTPUT
C0:
C1:
C2:
C3:
$\mathcal{A}_o^{0,1}(x,y)$
$\mathcal{A}_o^{0,2}(x,y)$
$\mathcal{A}_o^{0,3}(x,y)$
T

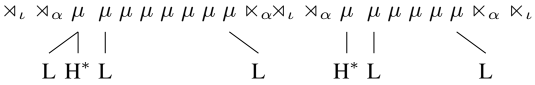
Figure 6: Melodic transduction of intonation in Seoul Korean

3.3 Tokyo Japanese

3.3.1 Basic intonational pattern

Tokyo Japanese is a lexical pitch accent language (Beckman and Pierrehumbert, 1986), where tones are lexically specified for particular moras, while other tones are defined in the phrase-level. The typical intonational pattern in Tokyo Japanese is a rising pitch pattern at the beginning of an Accentual Phrase (AP), which depends on where the lexical pitch accent H*L is realized. The actual f0 contour of a Japanese declarative for (4) is shown in Figure 9.

(4)


$\rtimes_\iota$ $\rtimes_\alpha$ μ μ μ μ μ μ μ μ $\ltimes_\alpha\rtimes_\iota$ $\rtimes_\alpha$ μ μ μ μ μ μ $\ltimes_\alpha$ $\ltimes_\iota$
L H* L
L
H* L
L

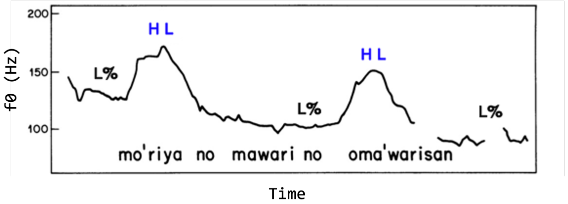


Figure 9: An f0 contour for a declarative intonation in Tokyo Japanese, extracted from Beckman and Pierrehumbert (1986).

When the first syllable of the first lexical item in an AP is *accented*, H*L is associated to the first mora of the accented syllable, with H* realized on the first mora and L on the second. This realization prevents an L% boundary tone and a phrasal H tone from associating with the first and second moras of the AP. Instead, the L% boundary tone of the preceding AP is linked to its final mora rather than the first mora of the current AP.

When the first syllable of the first lexical item in an AP *unaccented* (e.g., omáwarisan), a phrasal H tone is usually linked to the second sonorant mora and L% boundary tone of the preceding AP is associated to the first mora of the following AP.

Lastly, L% boundary tone is inserted at the beginning of the utterance as a whole. A postlexical rule deletes all accents after the first accent in an AP, which is known as deaccentuation.

3.3.2 Melodic transduction

Step 1: Copying The input is a string that consists of moras (μ, μ^*) and boundaries ($\rtimes_\iota/\ltimes_\iota$, $\rtimes_\alpha/\ltimes_\alpha$), as defined below. The outputs are five *copies* of the input, as shown in Figure 5. For C0, everything in the input is copied such that moras and boundaries in the output are interpreted the same as those in the input.

$$\begin{aligned}
\mu_o^0(x) &= \mu_i(x) & \mu_o^{*0}(x) &= \mu_i^*(x) \\
\ltimes_{\alpha_o}^0(x) &= \ltimes_{\iota_i}^0(x) & \ltimes_{\iota_o}^0(x) &= \ltimes_{\iota_i}^0(x) \\
\rtimes_{\alpha_o}^0(x) &= \rtimes_{\alpha_i}^0(x) & \rtimes_{\iota_o}^0(x) &= \rtimes_{\iota_i}^0(x) \\
H_o^{*1}(x) &= \mu_i^*(x) & L_o^2(x) &= \mu_i^*(x) \\
T_o^3(x) &= \rtimes_{\alpha_i}(x) & T_o^4(x) &= \rtimes_{\iota_i}(x) \wedge \ltimes_{\alpha_i}(x)
\end{aligned}$$

In C1 and C2, the HL lexical pitch accents ($H_o^{*1}(x)$ and $L_o^2(x)$) in the output are derived directly from the starred moras ($\mu_i^*(x)$) in the input, as they are lexically specified. This allows the actual HL tones to be computed in the output without creating unspecified tone slots like T. In C3 and C4, phrasal tones ($T_o^3(x)$) are derived from the left edge of an AP boundary ($\rtimes_{\alpha_i}(x)$), while boundary tones ($T_o^4(x)$) are derived from the left edge of an IP boundary ($\rtimes_{\iota_i}(x)$) or the right edge of an AP boundary ($\ltimes_{\alpha_i}(x)$). This direct mapping of input moras to lexical pitch accents and unspecified tones to post-lexical tones reflects Tokyo Japanese's pitch accent patterns.

Step 2: Tone-TBU association The tones in the melodic tiers (C1-C4) are associated with moras in the segmental tier (C0). For lexical pitch accents in the last AP, only the first pitch accent sequence (H^* in C1 and L in C2) is realized, while others are deaccented. This association is defined by

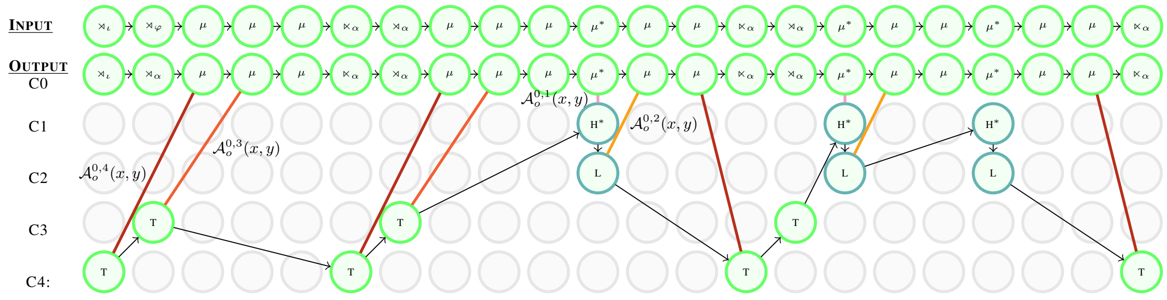

INPUT
OUTPUT
C0
C1
C2
C3
C4:
$\mathcal{A}_o^{0,1}(x,y)$
$\mathcal{A}_o^{0,2}(x,y)$
$\mathcal{A}_o^{0,3}(x,y)$
$\mathcal{A}_o^{0,4}(x,y)$
H*
L
T

Figure 8: Melodic transduction of Tokyo Japanese intonation.

$\mathcal{A}_o^{0,1}(x,y)$, linking the first starred mora after the left edge of an AP boundary with the H^* using the p^* function. Similarly, $\mathcal{A}_o^{0,2}(x,y)$ links L to the next mora. Subsequent pitch accent sequences in the last AP are not associated with their TBUs. $\mathcal{A}_o^{0,3}(x,y)$ associates the phrasal tones with the second mora in an AP only when not followed by a lexical pitch accent. Therefore, if the following elements are the lexical pitch accents, the phrasal tones cannot be realized. As for the boundary tones, $\mathcal{A}_{\text{o}}{}^{0,4}(x,y)$ associates the boundary tones with the first mora in an AP or with the last mora of the preceding AP or the final AP.

$$\mathcal{A}_o^{0,1}(x,y) = \mu_i^*(x) \wedge \rtimes_{\alpha_i}(y) \wedge y \approx p^*(x)$$
$$\mathcal{A}_o^{0,2}(x,y) = \mu_i(x) \wedge \rtimes_{\alpha_i}(y) \wedge y \approx p^*(x)$$
$$\mathcal{A}_o^{0,3}(x,y) = \mu_i(x) \wedge (\rtimes_{\alpha_i}(y) \wedge y \approx s(s(x))) \wedge \neg(\mu^*(y) \wedge y \approx s(x))$$
$$\mathcal{A}_o^{0,4}(x,y) = \mu_i(x) \wedge (\rtimes_{\iota_i}(y) \vee \rtimes_{\alpha_i}(y) \wedge y \approx p(p(x))) \vee (\ltimes_{\alpha_i}(y) \wedge y \approx s(x))$$

3.3.3 Declarative meaning transduction

After the melodic transduction, the unspecified post-lexical tones (Ts) are filled with Hs and Ls for the declarative in Tokyo Japanese in Figure 10. Note that the lexical pitch accents are already filled with H^* and L. The input signatures are $\{\mu, \mu^*, \ltimes_\alpha, \ltimes_\iota, \rtimes_\alpha, \rtimes_\iota, T, H^*, L\}$ and the output signatures are $\{\mu, \mu^*, \ltimes_\alpha, \ltimes_\iota, \rtimes_\alpha, \rtimes_\iota, H^*, H, L\}$. The formula is as follows: $L_o(x) = T_i(x)$.

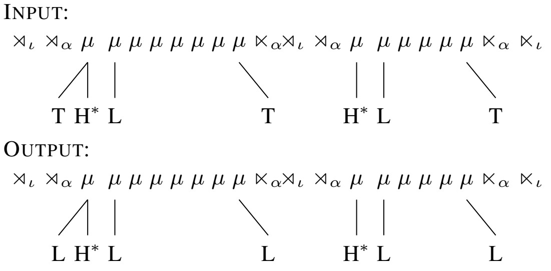

INPUT:
⋊ι ⋊α μ μ μ μ μ μ μ μ μ ⋉α⋊ι ⋊α μ μ μ μ μ μ ⋉α ⋉ι
T H* L T H* L T
OUTPUT:
⋊ι ⋊α μ μ μ μ μ μ μ μ μ ⋉α⋊ι ⋊α μ μ μ μ μ μ ⋉α ⋉ι
L H* L L H* L L

Figure 10: Declarative meaning transduction of Tokyo Japanese intonation.

3.3.4 Summary

Results showed that the intonational pattern in Tokyo Japanese can be defined using a QF logical interpretation of a prosodic structure. Unlike the post-lexical (head-prominence and edge-prominence) intonational patterns in American English and Seoul Korean, copying starred moras *directly* to specified tones—H* and L—was able to capture the lexically specified pitch accent in Tokyo Japanese. Also, copying boundaries was able to capture the realization of post-lexical (phrasal) tones. This process reflects the typical initial rising pitch in an AP in Tokyo Japanese. Even with deaccentuation, where only the first lexical pitch accent in an AP is realized, tone-TBU associations were defined *locally* without quantifiers, by making reference to tier-based representation.

4 Discussion

By defining the intonational structure as a QF logical interpretation of a metrical and prosodic structure that are ISL, we were able to create an *intonational theory* that is restrictive enough to characterize different intonational patterns.

From the typological view of intonation, the head-prominence intonational pattern in American English was defined with the copies of both starred syllables (i.e., heads) and boundaries, whereas the edge-prominence pattern in Seoul Korean was defined with the copies of only boundaries (i.e., edges). The lexical pitch accent pattern in Tokyo Japanese was defined with both copies of starred moras for the lexical pitch accent and copies of phrasal boundaries for the post-lexical tones.

This suggests that the prosodic elements in the input strings are not realized the same way, but the way they are logically interpreted leads to the characterization of different metrical and prosodic realizations in intonation.

Crucially, the computational nature of intonational tone-TBU association patterns found to be characterized as QF logical interpretations. As for the Melodic Transduction, the tone-TBU associations in both American English and Seoul Korean were analyzed in a strictly local manner, without the need of quantifiers. Even in the case of Tokyo Japanese, where prosodic elements like starred TBUs and boundaries may appear non-local, the QF logical interpretations are achieved by preserving the input order in the output (Chandlee and Jardine, 2019b) and using tier-based predecessor and successor functions (p^*, s^*). Furthermore, the use of these starred ordering functions captures the hierarchical structure of TBUs, reflecting their relative prominence, in line with the AM theory's view. Even within the class of QF logical interpretations, typological distinctions can be observed (Danis, 2025). The intonational patterns of Tokyo Japanese are found to be more complex, requiring the use of p^* and s^*, whereas those of American English and Seoul Korean can be captured without using such functions.

As for the Declarative Transduction, at least for American English, Seoul Korean, and Tokyo Japanese, H and L sequences were defined using FO logic without quantifiers. Notably, no case required even-numbered starred syllables to be H tones. This result can be extended to Question Transduction with similar tonal sequence except for an H boundary tone at the end of an IP. This QF logical characterization confirmed that intonational patterns are also ISL functions like most of other phonological mappings within the regular upper bound of phonology (Chandlee, 2014; Chandlee and Jardine, 2019b; Chandlee and Lindell, to appear).

Based on these results, we may able to ask several questions to predict the intonational patterns: 1) what kind of prosodic elements are being copied in the output? Is it a head of a constituent? Is it a phrasal boundary? Or are they both?; 2) when are the tones specified during the derivation from the input to the output? Is it directly specified from the input to the output in a melodic transduction? Or is it specified during the meaning transduction? These questions can provide valuable predictions of possible intonational patterns in the typology.

Further research is needed to generalize the locality of intonational patterns by examining more languages within the same intonational categories. For instance, Spanish is another head-prominence intonational language (Beckman et al., 2002), where the stressed syllable receives pitch accents (e.g., H*, L*+H) within an ip, and boundary tones (L%, H%) are realized at the end of an IP. The intonational pattern in Spanish may possibly seem to function similarly to that in American English, as the heads of constituents serve as main prosodic elements. In contrast, French is known for marking prominence at the edges of an AP (/LHiLH*/ (Jun and Fougeron, 2000), where the phrase-final H* on the last full vowel signals the edge of an AP, while the initial accent Hi is optionally realized. Boundary tones (H%, L%) are realized on the final syllable of an IP. The phrase-final edge-prominence properties in French can be compared to those in other edge-prominence languages like Seoul Korean.

As for lexical pitch accent patterns, Lekeitio Basque may exhibit similar patterns as in Tokyo Japanese. That is, in Lekeitio Basque, a H*+L lexical pitch accent is realized in an AP and a %L boundary tone is realized on the first syllable of an AP (Elordieta, 1998). An IP begins and ends with boundary tones (L%, H%). Due to the absence of a deaccentuation pattern, tonal computation in Lekeitio Basque may be less complex than in Tokyo Japanese. Likewise, we need further analyses on the intonational pattern of other languages to generalize our results that intonation is a QF logical interpretation of a metrical and prosodic structure that are defined locally. But in this way, we can provide a theory of intonation that makes restrictive predictions about the typology of intonation and measure the complexity of intonational structures.

5 Conclusion

The present study explored how the tone-TBU association patterns in intonation can be defined using a QF logical interpretation of a metrical and prosodic structure. Tones were construed as literal copies of prosodic elements, such as starred syllables or boundaries, and their associations with TBUs were defined locally without quantifiers. Head-and edge-prominence intonational patterns were QF metrical grids, whereas lexical pitch accent patterns were more complex. By defining intonation as a logical interpretation, we were able to understand the computational nature of intonation and predict the typology of intonation, contributing the theory of intonational and computational phonology.

References

Mary E. Beckman, Manuel Díaz-Campos, Joanne T. McGory, and Terrell A. Morgan. 2002. Intonation across spanish, in the tones and break indices framework. In Carlos Gussenhoven and Natasha Warner, editors, *Laboratory Phonology 7*, pages 545–576. Mouton de Gruyter, Berlin.

Mary E. Beckman and Janet B. Pierrehumbert. 1986. Intonational structure in japanese and english. *Phonology*, 3:255–309.

Jane Chandlee. 2014. *Strictly Local Phonological Processes*. Ph.D. thesis, University of Delaware.

Jane Chandlee and Jeffrey Heinz. 2018. Strict locality and phonological maps. *Linguistic Inquiry*, 49(1):23–60.

Jane Chandlee and Adam Jardine. 2019a. Autosegmental input strictly local functions. *Transactions of the Association for Computational Linguistics*, 7:157–168.

Jane Chandlee and Adam Jardine. 2019b. Quantifier-free least-fixed point functions for phonology. In *Proceedings of MOL 16*, pages 50–62.

Jane Chandlee and Adam Jardine. 2021. Input and output locality and representation. *Glossa: A Journal of General Linguistics*, 6(1):90.

Jane Chandlee and Steven Lindell. to appear. Logical perspectives on strictly local transformations. In Jeffrey Heinz, editor, *Doing Computational Phonology*. Oxford University Press.

Bruno Courcelle. 1994. Monadic second-order definable graph transductions: a survey. *Theoretical Computer Science*, 126:53–75.

Nick Danis. 2025. Logical transductions are not sufficient for notational equivalence. In *Proceedings of the Annual Meetings on Phonology*, volume 1. University of Massachusetts Amherst Libraries.

Jason Eisner. 1997. What constraints should ot allow? Paper presented at the Annual Meeting of the Linguistic Society of America, January 1997, in Chicago.

Gorka Elordieta. 1998. Intonation in a pitch accent variety of basque. *Anuario del Seminario de Filología Vasca "Julio de Urquijo"*, 32(2):511–569.

Herbert. B. Enderton. 2001. *A Mathematical Introduction to Logic*. Elsevier.

Joost Engelfriet and Hendrik Jan Hoogeboom. 2001. Mso definable string transductions and two-way finite-state transducers. *ACM Transactions on Computational Logic (TOCL)*, 2(2):216–254.

Emmanuel Filiot and Pierre-Alain Reynier. 2016. Transducers, logic and algebra for functions of finite words. *ACM SIGLOG News*, 3(3):4–19.

Jeffrey Heinz and William Idsardi. 2011. Sentence and word complexity. *Science*, 333(6040):295–297.

Brett Hyde. 2008. Alignment continued: distance-sensitivity, order-sensitivity, and the midpoint pathology. Manuscript, Washington University.

Adam Jardine. 2017. On the logical complexity of autosegmental representations. In *Proceedings of MOL 15*, pages 22–35.

Adam Jardine, Natalie Danis, and Luca Iacoponi. 2021. A formal investigation of q-theory in comparison to autosegmental representations. *Linguistic Inquiry*, 52(2):333–358.

C. Douglas Johnson. 1972. *Formal aspects of phonological description*. Mouton, The Hague.

Sun-Ah Jun. 2006. Intonational phonology of seoul korean revisited. *Japanese-Korean Linguistics*, 14:15–26.

Sun-Ah Jun and Cécile Fougeron. 2000. A phonological model of french intonation. In Merle Horne, editor, *Intonation: Analysis, modelling and technology*, pages 209–242. Springer Netherlands, Dordrecht.

Ronald M. Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

Nathan Koser, Christopher Oakden, and Adam Jardine. 2019. Tone association and output locality in non-linear structures. In *Supplemental Proceedings of the 2019 Annual Meeting on Phonology*. Linguistic Society of America.

Leonid Libkin. 2004. *Elements of Finite Model Theory*, volume 41. Springer, Heidelberg.

Janet B. Pierrehumbert. 1980. *The phonology and phonetics of English intonation*. Ph.D. thesis, Massachusetts Institute of Technology.

Kristine Strother-Garcia. 2019. *Using model theory in phonology: A novel characterization of syllable structure and syllabification*. Ph.D. thesis, University of Delaware.

Evidence of Hierarchically-Complex Syntactic Structure Within BERT's Word Representations

Mary Katie Kennedy
University of Southern California, Los Angeles, CA, USA
mkkenned@usc.edu

Abstract

Our research provides empirical support that LLM's contextualized word embeddings have captured deep and hierarchical syntactic structure. In 2019, Hewitt and Manning found evidence that LLMs have captured features of structural dependency parses within their word representations; we extend this work by deploying their methodology on sentence structures that are differentiated only in a constituency-based account like Minimalism rather than a dependency-based account. Our novel work creates a dataset containing several carefully selected sentence structures whose dependency parses are identical, but whose constituency trees differ due to to the size of the complement (vP versus TP versus CP). We find differences in the probe's predicted distances that can only be explained if the embeddings have indeed captured some Minimalist structural difference between these sentence types. The impact of our work helps to realize Linzen (2019)'s argument that linguists can further the study and understanding of LLMs and that the field of NLP provides novel tools for further linguistic research.

1 Introduction

Since the release of BERT (Devlin et al., 2019), much research has been done to test and expand the impressive performance of large language models. A subset of research interest lays in understanding what linguistic structures and knowledge these models have acquired (Jawahar et al., 2019; Belinkov and Glass, 2019; He et al., 2024; Waldis et al., 2024; Kallini et al., 2024), including syntactic (Clark et al., 2019; Chi et al., 2020; Kulmizev et al., 2020; Maudslay and Cotterell, 2021; Arps et al., 2022), morphological (Coleman, 2020; Anh et al., 2024), and semantic knowledge (Nikolaev and Padó, 2023; Kamath et al., 2024).

Our work extends this body of research by utilizing a probe method developed by (Hewitt and Manning, 2019), which finds that a dependency parse can be recovered solely from the contextualized vector embeddings of a pretrained language model like BERT (Devlin et al., 2019). We further these findings by deploying the probe on sentence structures whose dependency parse is invariant (i.e., the distance between a head and its dependent is always 1, see Section 2.1 for explanation), but whose hierarchical distances vary depending upon the size of a phrasal complement in a Minimalist constituency framework (see Section 3.2 for details). In doing so, we seek to discover whether large language models like BERT have captured the complex hierarchies and subsurface structures postulated by syntacticians in the Minimalist Program. This work thus follows in the research vein of Linzen (2019), who argues that linguists and NLP researchers stand in a unique position for collaboration to leverage the skills and tools of their respective fields to better understand, test, and develop the two bodies of research.

2 Background

2.1 Syntactic Theories

In the field of NLP, there are two main approaches to syntax that a researcher can utilize: a Dependency Grammar (DG) approach or a constituency grammar (CG), also known as a phrase-structure grammar. In brief, Dependency Grammar focuses more on the relationship between constituents without needing to represent a sentence's linearized word-order, making it popular for work on languages with freer word order (Müller, 2019).[1]

The core of the theory centers around the concept of *valence*, which indicates which words govern

[1] Various schools of thought in the theory have proposed different mechanisms to derive linear order from a dependency structure, including the idea that linear order is dictated by surface syntactic rules (Müller, 2019). The author of this approach, Ulrich Engel, published in 2014 in (Öhl, 2015), though the original source is in German.

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 271-286.
Eugene, Oregon, July 18-20, 2025

(1) Dependency Tree

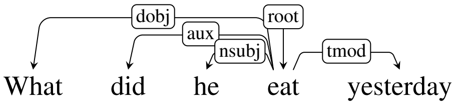


Figure 1: An example of the dependency tree for the sentence "What did he eat?" Note the flatter structure, the one-to-one mapping of words to nodes in the tree, and how each word has one and only incoming arc, excepting the root.

which words in a sentence. The governing word in a phrasal pair is considered the "head" and the governed word is its "dependent," sometimes called its "valence" (Müller, 2019). Each sentence will have one and only one "root," which is typically the matrix verb of the sentence, that will have no head itself. Thus, in a dependency tree, all words—except the root—will have one and only one incoming arc from its head. Though a word itself can head several other words, it itself can only be headed by one other word (see Example (1)).

On the other hand, constituency grammars are popular amongst many syntacticians and linguists who have built theories off of the work of Chomsky and others who have refined various aspects of phrase-structure/constituency-based grammars (Chomsky, 1957, 1981, 1986, 1995). Phrase-structure grammars are based around X-bar theory and operations of *Merge* and *Move* (Chomsky, 1995) and their consequent traces (Chomsky, 1973; Fiengo, 1977) (such as question formations where "He ate chicken" transforms into "What did he eat?"). After all syntactic operations are applied and all relevant nodes have been moved and/or merged, the end result is the sentence's linearization, meaning the final locations of the words in the hierarchy should match what is actually uttered if the tree is read from left to right (see Example (2)). Constituency-based grammars (CGs) thus result in trees with deep and complex hierarchies wherein empty nodes must be inferred as the traces and remnants of previous operations.

Like DG, many constituency theories incorporate the concept of valence, albeit with some modifications. Some of Chomsky's earlier work in the theory of Government-Binding (Chomsky, 1981) stipulates that certain categories (particularly the lexical categories of Verb, Noun, Adjective/Adverb, and Preposition in addition to the functional category of Tense) head/govern/dominate other constituents.[2] Later theories (Chomsky, 1995) refined this by defining specific operations, such as Merge, where the head element provides the properties of the combined result (e.g., Verb *eat* + Noun *chicken* = VerbPhrase *eat chicken*, not NounPhrase *eat chicken*), and which enables the recursive feature of language (e.g., "The old lady swallowed a fly that was then caught by a spider she later swallowed that was...."), thus allowing for infinite embeddings.

(2) Constituency Tree

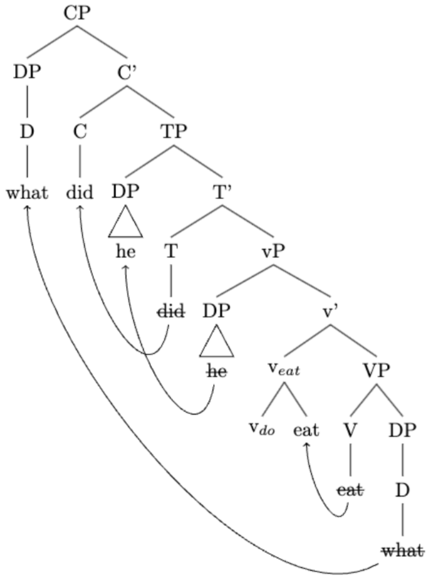


Figure 2: An example of a constituency tree for the sentence "What did he eat?" Note the depth of the tree and the movement of elements.

In short, both theories postulate a primitive building operation that allows for the combination of two elements into a single, new element whose features are determined by the head word, enabling the recursive nature of language to appear. For DG, this is through the dependency relationship, which establishes the head; for CG, this is through the Merge operation, which assigns the features of the phrase by referring to the phrase's head. The core differences, meanwhile, can be summed up as:

1. Dependency Grammars use a one-to-one mapping between words and nodes in the tree. Constituency Grammars more often use a one-to-many mapping between nodes in the tree,

[2] A constituent *A* can govern another constituent *C* iff *C* does not govern *A*, and there is no intervening element *B* that governs *A* but not *C*.

postulating branches and nodes that are not overtly present in the spell-out.

2. Dependency Grammars root at the verb. In Constituency Grammars, generally the Complementizer Phrase (CP) or Tense Phrase (TP)[3] exists as the highest level, though it is true that all sentences must have a verb in order to valid.

3. Structurally, Dependency Grammars do not distinguish between a head's arguments (e.g., the subject or object of a verb) and its adjuncts (e.g., modifiers, such as an adverb or prepositional phrase modifying the verb). The difference is left to the dependency label, but the structure remains changed. In contrast, Constituency Grammars, particularly Minimalism, structurally distinguish between the two, and even between argument types.

4. Dependency Grammars opt for reduced, flatter, more horizontal representation of word-to-word relationships. Constituency Grammars opt for a more hierarchically complex, vertically-organized representation.

When syntax is leveraged in NLP, the framework adopted tends to be DG rather than CG (compare 14,900 ACL papers on Dependency Grammar as opposed to only 3,630 on Constituency Grammar). There are several reasons for this: DG's trees are simpler (nodes are in a one-to-one relationship with words), DG is more static (dependencies are assessed in-situ, meaning one needs not be concerned whether or not an element moved to its location or base-generated there), DG utilizes flatter representations (because elements are assessed in-situ, there is no need to postulate more complex and empty hierarchies that might explain how or why the word is currently where it is), and its simplicity and avoidance of contentious theoretical debates—such as those in Minimalism—allow for faster and more consistent inter-annotator agreement.

The DG framework is appealing to many in NLP as it is relatively easy to learn and its compact and efficient representation has proven to be salutary to downstream tasks, such as question-answering, relation extraction, summary (de Marneffe et al., 2006), spam detection (Milner, 2024), sentiment analysis (Liang et al., 2021), sentence classification and matching as well as sequence labeling and machine translation (Zhang et al., 2021), and more. However, the theory fails to capture linear order, nor does it explain the patterns and restrictions that form licit sentences and their interpretations, and it furthermore entirely skirts the issues of the deep and complex hierarchies that have been argued for in Minimalism. In this vein, we seek to investigate to what extent LLMs have captured the deeper and more complex syntactic structures proposed by constituency grammar frameworks, such as Minimalism.

2.2 Probes

Since LLMs took the world by storm with their impressive performance in multiple language tasks, researchers have sought to understand what linguistic properties LLMs have actually acquired. A popular method is the probe method, first proposed by Shi et al. (2016), which used the embeddings from neural machine translation encoders to train a logistic regression classifier in order to identify what syntactic features were acquired by the models. This field of research and these probe models are not concerned with improving state-of-the-art performance; rather, they seek to investigate, or "probe", what latent linguistic features a language model has acquired.

The tasks specified by probes depend on the linguistic feature under investigation (e.g., semantics, syntax, etc.), but often utilize a pretrained language model's latent features, such as their vector representations (Conneau et al., 2018; Jawahar et al., 2019; Tenney et al., 2019b,a; Starace et al., 2023) or attention mechanisms (Clark et al., 2019; Manning et al., 2020).

One form of structural probe, developed by Hewitt and Manning (2019), found that the pretrained contextualized embeddings of BERT (Devlin et al., 2019) and ELMo (Peters et al., 2018) could be used to recover dependency trees from those vector representation of words. To find this, Hewitt and Manning trained a linear transformation matrix to take the contextualized word embeddings and project them into a subspace where the squared Euclidean distance between word nodes ultimately recovers a dependency parse. That is to say, their probe's training objective was to learn to map words' contextualized embeddings to new positions within a subspace where the probe's predicted squared Euclidean distance between each head and its de-

[3] Some languages do not include tense, like Chinese, and so the top level is often represented as IP for Inflectional Phrase.

pendent is approximately 1.[4]

While Hewitt and Manning (2019) and others (Chi et al., 2020; Kulmizev et al., 2020; Müller-Eberstein et al., 2022; Eisape et al., 2022) have found evidence that dependency structures are encoded within the contextualized vector representations, it remains unclear whether LLMs have acquired the deep, hierarchically-complex structures of constituency grammars such as those proposed in the Minimalist framework. To this end, we utilize the structural probe of Hewitt and Manning and test sentence types whose hierarchical distance varies in a constituency/Minimalist account, but whose head-dependency distance does not vary in a Dependency Grammar account. If the probe is sensitive to the nuances of a constituency account, this indicates that not only have the language models captured something of the hierarchically complex and subsurface structures of Minimalism, but that a probe trained only to recover a dependency parses is capturing constituency syntax for free.

3 Methods

Our work is not the first research to probe at constituencies (Tenney et al., 2019b; Arps et al., 2022; Kallini et al., 2024). However, these previous methods either focus solely at the phrase-level by seeking to train a probe to recover a phrase's boundaries (Tenney et al., 2019b; Kallini et al., 2024) or by training on the English Penn Treebank for their probe (Arps et al., 2022). While constituency trees represented in the English Penn Treebank (Marcus et al., 1993) are deeper than their equivalent dependency trees, they do not adhere to the binary branching requirement postulated in Minimalism and do not capture Merge and Move operations. As such, the representations are not as rich nor as complex as those which have been posited in the Minimalist constituency framework.

For this reason, we opt for the novel approach of utilizing the original Hewitt and Manning (2019) structural probe that was trained to recover dependency trees to probe for variations in constituency hierarchies. To that end, our stimuli involve sentences wherein the distance between a head and its dependent is invariant in a DG account, but whose hierarchical distance depends upon the sentence structure as captured in the Minimalist framework. The choice to probe for a dependency parse as opposed to a constituency in fact allows us to avoid several potential pitfalls of constituency trees: namely that constituency trees make assumptions about the underlying structure and may predispose the probe to recover the constituency parses utilized in the training data rather than probing for a latent representation of constituency hierarchies as captured by the model.

3.1 Computational Model

The structural probe by (Hewitt and Manning, 2019) stipulates a model M that produces a sequence of vector representations $h^{l}_{1:n}$ from an input sequence of n words $w^{l}_{1:n}$ where l identifies the sentence. A linear transformation $B \in \mathbb{R}^{k \times n}$ parameterizes the parse tree-encoding distances:

$$d_B(\mathbf{h}^l_i, \mathbf{h}^l_j)^2 \quad = \quad (B(\mathbf{h}^l_i - \mathbf{h}^l_j))^T(B(\mathbf{h}^l_i - \mathbf{h}^l_j))$$

where i and j are the words in the sentence and where the matrix B is trained to reproduce the gold parse distances between each pair of words (w^l_i, w^l_j) in each sentence for all the sentences within the parsed training corpus T^l.[5] This training is accomplished through the gradient descent objective:

$$\min_B \sum_l \frac{1}{\mid s^l \mid^2} \sum_{i,j} \mid d_{T^l}(w^l_i, w^l_j) - d_B(h^l_i, h^l_j)^2 \mid$$

In doing so, the objective seeks to approximate the matrix that most closely reproduces distances that align with the gold-standard distances. $\mid s^l \mid$ is the length of the sentences, and the function normalizes using the square of the sentence's length since each sentence contains $\mid s^l \mid^2$ pairs of words.

Hewitt and Manning (2019) trained their structural probe using BERT-large (cased) with 1024 dimensionality for all 24 layers. The probe was trained with the objective of minimizing the L1 loss of the predicted squared distance with respect to the true distance (i.e., the distance between a head and its dependent should be 1; the distance between the dependent of a dependent of a head should be 2; and so on). They used Adam optimizer (Kingma and Ba, 2014) with an initialized learning rate of 0.001 with $\beta = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-2}$ and an epoch maximum of 40 or to convergence with a batch size of 20. Dev loss

[4] The specific mathematics and model information can be found in Section 3.1.

[5] The authors found that training on squared distances and using the square root to retrieve the final distance performed better than using the direct distance. Hewitt and Manning (2019) left the possible reasoning for this for future work.

was calculated at each epoch; if the dev loss was not a new low for the model, the optimizer was reset with an initial learning rate multiplied by 0.1. The probe was implemented using DyNet (Neubig et al., 2017) and PyTorch (Paszke et al., 2019).

Probe evaluation was based on how closely the predicted distances between word pairs align with the gold parse structures, which were created by converting the constituency trees from the English Penn Treebank (Marcus et al., 1993) into dependency parses.[6] To measure this, the authors calculated the minimum spanning tree for each sentence's predicted distances and scored the undirected, unlabeled attachment score (UUAS), which merely measures whether or not the proper word-pairs are in a dependency relationship, ignoring the matter of directionality (which indicates which word is the head and which is the dependent in a head-dependent pair) and labels.

3.2 Linguistic Data

To probe whether vector embeddings encode the hierarchical distances captured by Minimalist constituency trees, we utilize the filler-gap dependencies that result from *wh*-question formation of sentences with embedded sentential complements (e.g., "What did she see [him eat __]"). By varying the size of the complement taken by the matrix verb and extracting out of that embedded complement, we can vary the constituency tree's hierarchical distance while keeping dependency distances constant.[7]

In traditional Minimalism, there is an accepted order to the hierarchy of phrases. At the highest level is the **complementizer phrase**, which introduces whether the clause is interrogative or declarative; under the CP is the **tense phrase**, which hosts tense information; the TP nests a **verb phrase**, which can further be subdivided into a small verb phrase (vP) also known as a **voice phrase** that takes a VP complement itself (Adger, 2003).

Different verbs can vary in the type and size of the complement they can take. At the largest level, a verb can take an entire finite clause as its complement (see Example (3)). Examples of such verbs include *think, believe, suspect, claim,* etc., which can all optionally include an overt complementizer like *that* or *who.*

(3) Full CP Complement

a. I think [$_{\mathrm{CP}}$ (that) he ate the chicken]

The next smallest complement size is a non-finite complement. The easiest one to discuss is the infinitive complement in sentences known as *exceptionally case marked* (ECM) (see Example (4)), which include matrix verbs that take TP complements (Adger, 2003). ECMs are called exceptionally marked because the subject of the embedded clause receives its accusative case (rather than the typical nominative case) from the matrix verb.

(4) ECM TP Complement

a. I expect [$_{\mathrm{TP}}$ him to eat the chicken]

Another small subset of verbs in English allow for phrasal complementation. This subset of verbs include causatives (e.g., *make, let*) and perception verbs (e.g., *see, hear, watch, feel*) that take bare infinitives (see Example (5)). We follow in the steps of (Sheehan and Cyrino, 2023) in analyzing these as vPs, which we dub "bare vPs" to emphasize that the nonfiniteness is not overtly realized with an infinitival *to* as it is in ECMs.

(5) Bare Infinitive Complement

a. I saw [$_{\mathrm{TP/vP}}$ him eat the chicken]

For our experimental design, we specifically needed sentence structures in which the dependency parse remained consistent, but the constituency parse yielded differing distances between two elements. For this reason, we leveraged the ability for verbs to take complements of differing sizes (vP, TP, and CP) and created *wh*-questions (e.g., *what did you see him eat/what did you expect him to eat/what did you think he ate*). *Wh*-question structure was specifically selected as the distance between the embedded verbal head (e.g., *eat*) and

[6] It is important to note here that the constituency trees of the Penn Treebank are not the binary branch trees with Merge and Move operations as postulated in Minimalism.

[7] While our experiment utilizes filler-gap dependencies, our probe method can be applied to any sentence structure types whose constituency tree varies but whose relevant dependency parse does not. Hewitt and Manning (2019) probe's training objective allows for flexibility in possible Minimalist structures. Its training objective is such that a parent-child relationship between a head and its dependent should return a distance of approximately 1, while a "grandparent"-child relationship (the dependent of a dependent of a head) should return a distance of approximately 2, and so on. Using this feature, Kennedy (2025) deploys our probing method on declarative Subject-Raising and Subject-Control constructions—the former of which is argued to take a smaller TP complement compared the latter's larger CP complement—and finds that the predicted Euclidean distance between matrix elements and embedded elements are larger in the Subject-Control condition despite the two structures having identical dependency parses.

its dependent (e.g., *what*) is consistently 1 in all conditions; however, in a Minimalist account, the hierarchical distance between embedded verb and its moved object depends on the size of the complement taken (vP, TP, or CP). For visualization, see the trees in Appendix B, Examples (6)-(9).

To add further complexity, two more sets of sentences were constructed that took advantage of the recursive property by creating sets for double-nested ECMs (e.g., *What did you expect her to want him to eat*) and double-nested full-CP complements (e.g., *What did you believe she suspected he ate*).

Using only pronouns for the subjects, the minimum linear distance (meaning the number of intervening words) between the extracted *wh*-constituent and the embedded verb ranged from 5 (bare vP and single CP) to 6 (single TP) to 7 (double CP) to 9 (double TP). Because the sentences could not be started at identical linear distances due to the presence of necessary words (such as *to* in ECMs), the linear distance was increased incrementally through the change of a pronoun (e.g., *you*) to a nominal phrase (e.g., *the professor*) to a modified nominal phrase (e.g., *the brilliant professor*) to a possessive nominal (e.g., *the brilliant professor's friend*) to the inclusion of an adverb.[8]

Using the above schema, we created a total of 18,252 carefully constructed sentences that strictly conformed to one of the five specific syntactic constructions that are well-accepted in traditional syntax as demonstrating different syntactic hierarchies.

4 Experiment

For our experiment, we used the best-performing pretrained probe from Hewitt and Manning (2019), which they found to be the probe for Layer 16 and which they released and made publicly available on their Github.[9] Our methodology sought to discover whether the probe's predicted squared Euclidean distances between head-dependent words were sensitive to hierarchical depth as postulated in a Minimalist framework. In a DG framework, the distance between a head and its dependent **should always be 1** across our five conditions. However, in a Minimalist account, the size of the complements (vP, TP, CP, TP-TP, and CP-CP) yields **longer and longer hierarchical distances** between the moved *wh*-object and the embedded verbs.

The contextualized embedding representations of our 18,252 sentences were fed into the pretrained probe, and we extracted the squared Euclidean distances between the new projections of the *wh*-word and the embedded verb *if and only if* the minimum spanning tree correctly established a head-dependent relationship between moved *wh*-word (the first word) and the in-situ embedded verb (the last word). As our experimental design rests upon comparing the predicted squared Euclidean distance of a *dependency* probe when given sentences whose structures vary only in a *constituency* Minimalist account, we were only interested in sentences in which the probe correctly identified the head-dependent relationship because there is little point in comparing the predicted dependency distances of an incorrect dependency parse.[10]

4.1 Predictions

The structural probe was trained only to recover latent dependency representations captured by the pretrained BERT model. Thus, the probe has no specific or overt reason to show sensitivity to constituency-based distances. If the probe is sensitive only to dependency representations, then the five conditions should show no difference in distances predicted by the model. Alternatively, it is possible that the contextualized vector representations have captured Minimalist-like syntax, but that the dependency-trained probe is insensitive to such features.

The more interesting outcome, however, would be if the model's predicted distances *are* affected by the constituency distances. If predicted distances are reflective of an influence of constituency distances, this would suggest 1. that the model itself captures some representation of Minimalist-like constituency in addition to dependency, and 2. that the dependency representations themselves are sensitive to constituency differences. Such findings would have implications for modeling this distinction in the theory of Dependency Grammar.

If it is found that the probe is able to pick up on constituency hierarchies, then we would anticipate that embedded verbs with CP complements should have the highest predicted distance as it has the highest number of hierarchical nodes between the

[8] For more detail on our dataset creation, see Appendix A.

[9] https://github.com/john-hewitt/structural-probes

[10] While the fail cases are of interest for further research and investigation, for our current purposes, robust analysis could only be conducted when the probe achieved its trained gold parse.

extracted *wh*-object and the embedded verb within the constituency tree. ECM verbs that take TP complements and perception verbs that take either bare vP complements should trail behind this.

4.1.1 Dependency vs Constituency for Probes

As mentioned, our probe is intentionally trained to recover dependency parses as opposed to constituency trees. While it may seem intuitive to utilize a probe trained to recover constituency trees like Arps et al. (2022), we argue that using a dependency probe for Minimalist constituency structures actually has several advantages.

The logic behind linguistic probes is that in order for them to be successful, the embedding representation (or attention scores for some probes) must encode some feature(s) of that linguistic phenomenon in order for the probe to be able to solve the task. However, one critique of probing methods is the concern that the probe may simply be learning the linguistic task rather than revealing latent features encoded within the representation (Hewitt and Liang, 2019). Our stance is that using a dependency probe to test for constituency-based hierarchical distances avoids this possible liability.

The Hewitt and Manning (2019) probe is trained to recover *only* head-dependency relationships such that the distance between a head and its dependent is approximately 1. While the constituency trees for our stimuli will vary in the number of intervening nodes between the extracted *wh-* word and its verb (with the hierarchical distance being largest with a CP complement followed by a TP complement followed by a vP complement), the dependency parses have an invariant distance of 1 (see examples (6)–(9) in Appendix B for visualization). Because the probe *isn't* trained to predict a syntactic size difference between the complement types, the predicted squared Euclidean distances shouldn't vary *unless the probe is picking up on some additional linguistic feature within the vector representation*. The training objective is naive to a difference in the complement sizesm, and because of this, the training objective cannot bias the probe to output a desired structural difference. Therefore, if the probe's distances *do* vary in theoretically-predicted ways, we can have a greater confidence in significant results that constituency hierarchical distances are captured within vector representations and that such representations are utilized to some extent to recover dependency parses. In this regard, our methodology helps to address issues raised by Maudslay et al. (2020) that an overly powerful probe blurs the line between probe and parser.

The second benefit of using a dependency-trained probe as opposed to a constituency-trained probe is that we can avoid biasing certain debated syntactic analyses. Kuznetsov and Gurevych (2020) finds that the linguistic formalism utilized can impact how a probe performs, both in its accuracy scores and in the means through which it makes predictions (e.g., which attention layers are utilized). A probe that seeks to recover constituency parses will inevitably need to pick a "gold" standard tree that includes structure whose syntactic analysis varies even within the Minimalist framework.

For example, we mentioned how perception verbs are debated to take either a vP (Sheehan and Cyrino, 2023) or bare TP (Felser, 1998) complement. Were we to train a constituency probe, we would need to overtly pick one side of the argument and would include training data that reflects one analyses, thus risking biasing the probe towards that particular analysis. Dependency parses, meanwhile, are minimalistic (but not Minimalist) in that they make few theoretical assumptions with the most important being that there exists a dominance relationship between a head and its dependent. Using a probe trained for minimalistic dependency parses lets us to remain as theoretically-agnostic as possible within the general Minimalist framework and allows us to probe for models' representational differences as opposed to imposing debated syntactic structures upon the probe.

5 Results

Of the 18,252 sentences fed to the probe, 4,034 properly established a dependency relationship between the *wh*-word and the embedded verb.[11] A linear mixed effect model was then fit using the constituency hierarchical representation (EmbedType), the linear distance between the target words (LinDist), and the interaction of the two as predictors. EmbedType was a categorical predictor that included perception verbs (BareVP), singular ECMs (SingTP), singular CP complements (SingCP), double ECMs (DoubTP), and double CP complements (DoubCP), which were all simple coded with BareVP as the reference level. Linear distance was a discrete variable. A by-Verb (the

[11] As mentioned, overall probe performance on these edge-case sentences is not the focus of this research, but discussion can be found in Appendix D.

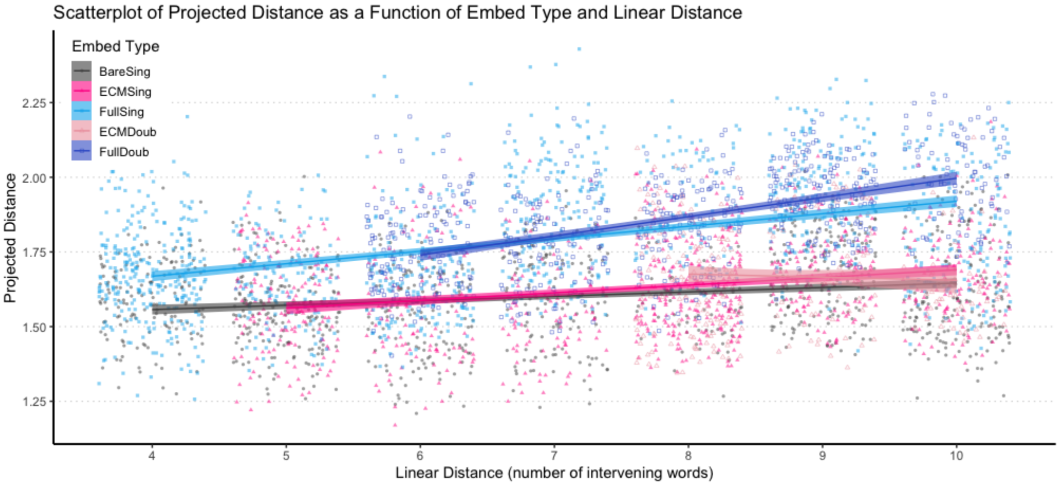


Figure 3: Scatterplot of projected distances as a function of linear distance (LinDist) and size of the verbal complement (EmbedType). There exists a stark difference between the larger CP complements and VP/TP complements. Statistical analysis reveals a significant difference between all conditions when considering their interactive effective with linear distance.

most deeply embedded verb; "eat" in our previous examples) uncorrelated random slope was added to the model.

In general, we can observe that as linear distance increases, so does the projected distance (see Figure 3). This is not surprising as it is well known that longer linear spans between dependencies tends to worsen performance as the number of intervening tokens are more likely to exceed that which is observed in training (Tenney et al., 2019b). More interesting is the clear divide in projected distances for the CP-levels versus the TP and vP levels.

The linear mixed effect model revealed significant main effects for singular TP and double CP embeddings (SingTP and DoubCP) compared to perception verb embeddings (BareVP) (see Table 1). That both SingTP and DoubCP reported projected distances that were significantly longer than the perception verb condition suggests that the probe is sensitive to constituency size.[12]

Additionally, increases in linear distance significantly corresponded to larger projected distances, though this was anticipated. Furthermore, significant interactions were found between linear distance and SingTP, linear distance and SingCP, and linear distance and DoubCP. The interaction between linear distance and DoubTP did not achieve significance, but that may be due to the notably fewer examples due to low UUAS performance.

Follow-up models were run on all categorical predictors (BareVP, SingTP, SingCP, DoubTP, DoubCP) to investigate interactions with linear distance. For all constructions, linear distance was a significant factor and the projected distances of all constructions, except DoubTP, increased with linear distance. This is expected as the greater linear distances between the two target words yielded poorer parse accuracy by the probe. That DoubTP does not conform to this behavior is likely due to it being a rare construction with few samples in our statistical analysis as the probe struggled to correctly establish the proper dependency relationship for this sentence structure.

6 Discussion & Conclusion

When linear distance is taken into account, a picture emerges in which the size of the complement (vP vs TP vs CP) is distinctly captured by the probe's correlatively larger projected distances (for further discussion, see Appendix C). These findings reveal to us several important conclusions:

1. The significant and correlative differences in projected distances between the different complement types suggest that pretrained models like BERT have learned representations that approximate in some capacity this hierarchical distinction between different complement sizes. Or, at the very least, it has picked up on

[12] That SingTP is significantly longer than BareVP but not DoubTP likely comes down to DoubTP having a much smaller sample size as this particular construction is more rare in natural data and yielded some of the lowest performance results by the probe.

Fixed Effects

Coefficient	$\hat{\beta}$	**SE($\hat{\beta}$)**	**t**	**df**	**p**
Intercept	1.462e+00	2.915e-02	8.458e+01	50.144	**2e-16**
SingTP	-1.692e-01	3.583e-02	9.217e+01	-4.723	**8.30e-06**
SingCP	-2.437e-02	2.521e-02	9.242e+01	-0.967	0.336258
DoubTP	-1.208e-01	9.775e-02	1.166e+03	-1.236	0.216882
DoubCP	-1.474e-01	4.091e-02	1.752e+02	-3.602	**0.000411**
LinDist	3.918e-02	2.399e-03	3.214e+03	16.332	**2e-16**
SingTP:LinDist	2.387e-02	3.383e-03	3.855e+03	7.055	**2.04e-12**
SingCP:LinDist	2.804e-02	2.595e-03	3.942e+03	10.805	**2e-16**
DoubTP:LinDist	1.893e-02	1.083e-02	2.703e+03	1.747	*0.080677*
DoubCP:LinDist	4.480e-02	4.361e-03	3.657e+03	10.275	**2e-16**

Random Effects

Group	**Term**	**Variance**	**Std.Dev**	**Corr.**			
Verb	Intercept	0.009815	0.09907				
	SingTP	0.030851	0.17564	-0.03			
	SingCP	0.039091	0.19772	0.10	0.82		
	DoubTP	0.008259	0.09088	-0.34	-0.30	-0.35	
	DoubCP	0.036986	0.19232	-0.24	0.72	0.82	-0.12
Residual		0.010048	0.10024				

Table 1: Number of observations: 4034. Groups: Verb (26). *P*-values/df calculated using the Satterthwaite approximation. Model formula: ProjDist Embed-Type*LinDist + (1 + EmbedType | Verb). Marginal *R2* = 0.2735, Conditional *R2* = 0.6487.

some quality of these constructions (e.g., finite vs non-finite) that corresponds to a greater or lesser extent with a distance in which finite constructions establish further distances from their moved object and their embedded verb when compared to non-finite counterparts.[13] This benefits the field of NLP by helping to better understand what qualities and features of languages these models have implicitly learned.

2. That a probe, specifically one trained only to recover dependencies, shows a sensitivity corresponding to a constituency-based analysis indicates to us that the theory of Dependency Grammar may have reason to specifically account for these relative distances. At the very least, we must postulate that this dependency probe is sensitive to finite constructions in that they show longer dependencies compared to non-finite constructions. The possibility of needing to account for some nested hierarchy in Dependency Grammar has already been proposed in order to explain certain syntactic patterns (Müller, 2019).

3. If pretrained models have indeed implicitly learned constituency representations in some capacity (or some parallel measure), then it may be that for the purpose of further NLP work, we do not need to incorporate the far denser and more complex constituency-based grammatical representations. While such theory has advantages and we find support for its analysis as a means to explain our data, the fact remains that the representations are extensive, requiring many branches, movement, empty nodes, and redundancies. The structures, though detailed, are too cumbersome to be easily implemented in NLP architectures, nor is it as accessible of a theory to utilize; scientists from other disciplines will have an easier time quickly learning and easily representing a dependency structure rather than a phrase structure. And if the dependency representations themselves are already affected by some constituency elements, then there may be less of an impetus to require computer scientists to learn an interesting and detailed but laborious representation when the nuances of the structures are already gotten for free in the models' geometries of their dependency representations.

The findings of this work have implications for the NLP field and the field of theoretical syntax. Not only does this work find evidence for the rich, subsurface syntax postulated by constituency theories such as Minimalism, but it furthermore finds evidence that LLMs are not only capable capturing generative Minimalist syntactic structures, but that they already do so to some extent. Our results also show support for the continuation of work like Müller (2019), who proposes utilizing nested hierarchies in Dependency Grammar to account for the structures captured by Minimalism and now by LLMs, too. Furthermore, the work teases as the possibility of utilizing LLMs for linguistic research. If these models are capturing theories postulated in syntax, might they not also be suitable as a means of testing theories when paired with human-based judgments? Already, our results suggest that BERT may favor (Sheehan and Cyrino, 2023)'s vP analysis over bare TP accounts as the probe's distances are significantly shorter than ECM's TP distances.

For the field of NLP, this provides evidence that the linguistic properties captured by LLMs are richer and more complex than previously realized, and that utilizing a dependency framework is still adequate as it appears that methods using dependencies are likely capturing constituency hierarchies for free. Overall, this work helps to realize Linzen (2019)'s claim that the skillsets and knowledge of

[13] Such coincidences already would be suspicious enough, and warrant further investigation to draw more conclusive interpretations.

the fields of NLP and Linguistics complement each other, and that the collaboration of two can help to further the respective fields.

Acknowledgements

I would like to thank Khalil Iskarous, Jon May, and Andrew Simpson for their informative discussions that have helped to enrich this research. I would also like to acknowledge and thank the reviewers whose feedback has lead to greater clarity within this project.

References

David Adger. 2003. *Core Minimalism*. Oxford University Press.

Dang Anh, Limor Raviv, and Lukas Galke. 2024. Morphology matters: Probing the cross-linguistic morphological generalization abilities of large language models through a wug test. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 177–188, Bangkok, Thailand. Association for Computational Linguistics.

David Arps, Younes Samih, Laura Kallmeyer, and Hassan Sajjad. 2022. Probing for constituency structure in neural language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6738–6757, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.

Noam Chomsky. 1957. *Syntactic Structures*. Mouton.

Noam Chomsky. 1973. Conditions on transformations. In *A Festschrift for Morris Halle*. Hole, Rinehard Winston.

Noam Chomsky. 1981. *Lectures on government and binding*. Foris Publications.

Noam Chomsky. 1986. *Knowledge of Language: Its Nature, Origin, and Use*. Praeger.

Noam Chomsky. 1995. *The Minimalist Program*. MIT Press, Cambridge, MA.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT‘s attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Haley Coleman. 2020. This is a BERT. now there are several of them. can they generalize to novel words? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 333–341, Online. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC’06)*, Genoa, Italy. European Language Resources Association (ELRA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tiwalayo Eisape, Vineet Gangireddy, Roger Levy, and Yoon Kim. 2022. Probing for incremental parse states in autoregressive language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2801–2813, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Claudia Felser. 1998. Perception and control: a minimalist analysis of english direct perception complements. *Journal of Linguistics*, 34(2):351–385.

Robert Fiengo. 1977. On trace theory. *Linguistic Inquiry*, 8:35–61.

Linyang He, Peili Chen, Ercong Nie, Yuanning Li, and Jonathan R. Brennan. 2024. Decoding probing: Revealing internal linguistic structures in neural language models using minimal pairs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4488–4497, Torino, Italia. ELRA and ICCL.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings* *of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Richard Hudson. 1984. *Word Grammar*. Blackwell Publishers.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Julie Kallini, Isabel Papadimitriou, Richard Futrell, Kyle Mahowald, and Christopher Potts. 2024. Mission: Impossible language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14691–14714, Bangkok, Thailand. Association for Computational Linguistics.

Gaurav Kamath, Sebastian Schuster, Sowmya Vajjala, and Siva Reddy. 2024. Scope ambiguities in large language models. *Transactions of the Association for Computational Linguistics*, 12:738–754.

Mary Katie Kennedy. 2025. Evidence of generative syntax in LLMs. In *The SIGNLL Conference on Computational Natural Language Learning*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Artur Kulmizev, Vinit Ravishankar, Mostafa Abdou, and Joakim Nivre. 2020. Do neural language models show preferences for syntactic formalisms? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4077–4091, Online. Association for Computational Linguistics.

Ilia Kuznetsov and Iryna Gurevych. 2020. A matter of framing: The impact of linguistic formalism on probing results. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 171–182, Online. Association for Computational Linguistics.

Yunlong Liang, Fandong Meng, Jinchao Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2021. A dependency syntactic knowledge augmented interactive architecture for end-to-end aspect-based sentiment analysis. *Neurocomputing*, 454:291–302.

Tal Linzen. 2019. What can linguistics and deep learning contribute to each other? response to pater. *Language*, 95(1):e99–e108. Publisher Copyright: © 2019, Linguistic Society of America. All rights reserved.

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, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Rowan Hall Maudslay and Ryan Cotterell. 2021. Do syntactic probes probe syntax? experiments with jabberwocky probing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 124–131, Online. Association for Computational Linguistics.

Rowan Hall Maudslay, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell. 2020. A tale of a probe and a parser. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7389–7395, Online. Association for Computational Linguistics.

Michael Milner, Helen Baron. 2024. Establishing an optimal online phishing detection method: Evaluating topological nlp transformers on text message data. *Journal of Data Science and Intelligent Systems*, 2:37–45.

Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2022. Probing for labeled dependency trees. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7711–7726, Dublin, Ireland. Association for Computational Linguistics.

Stefan Müller. 2019. *Superseded: Grammatical theory*. Number 1 in Textbooks in Language Sciences. Language Science Press, Berlin.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, and Pengcheng Yin. 2017. Dynet: The dynamic neural network toolkit.

Dmitry Nikolaev and Sebastian Padó. 2023. Investigating semantic subspaces of transformer sentence embeddings through linear structural probing. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 142–154, Singapore. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *ArXiv*, abs/1802.05365.

Michelle Sheehan and Sonia Cyrino. 2023. Restrictions on Long Passives in English and Brazilian Portuguese: A Phase-Based Account. *Linguistic Inquiry*, pages 1–35.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas. Association for Computational Linguistics.

Giulio Starace, Konstantinos Papakostas, Rochelle Choenni, Apostolos Panagiotopoulos, Matteo Rosati, Alina Leidinger, and Ekaterina Shutova. 2023. Probing LLMs for joint encoding of linguistic categories. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7158–7179, Singapore. Association for Computational Linguistics.

Stanley Starosta. 1988. *The Case for Lexicase: An Outline of Lexicase Grammatical Theory*. Pinter.

Stanley Starosta. 1997. *Reconnecting Language : Morphology and Syntax in Functional Perspectives*, chapter Control in constrained dependency grammar. John Benjamins Publishing Company.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. *Preprint*, arXiv:1905.06316.

Andreas Waldis, Yotam Perlitz, Leshem Choshen, Yufang Hou, and Iryna Gurevych. 2024. Holmes a benchmark to assess the linguistic competence of language models. *Transactions of the Association for Computational Linguistics*, 12:1616–1647.

Meishan Zhang, Zhenghua Li, Guohong Fu, and Min Zhang. 2021. Dependency-based syntax-aware word representations. *Artificial Intelligence*, 292:103427.

Peter Öhl. 2015. Jörg hagemann sven staffeldt (hg.). 2014. syntaxtheorien. analysen im vergleich. *Zeitschrift für Rezensionen zur germanistischen Sprachwissenschaft*, 7(1-2):19–26.

A Stimuli

Our dataset utilized five structure conditions (Bare vP, Singular TP, Singular CP, Double TP, and Double CP). Our key verbs for the five conditions included:

1. Bare vP: see, hear, watch
2. Singular TP: require, allow, want
3. Singular CP: think, suspect, claim
4. Double TP: expect + {require, allow, want}
5. Double CP: believe + {think, suspect, claim}

Additionally, we varied the subjects for our sentences in order to vary the linear distance between the *wh*-word and the embedded verb. These subjects included:

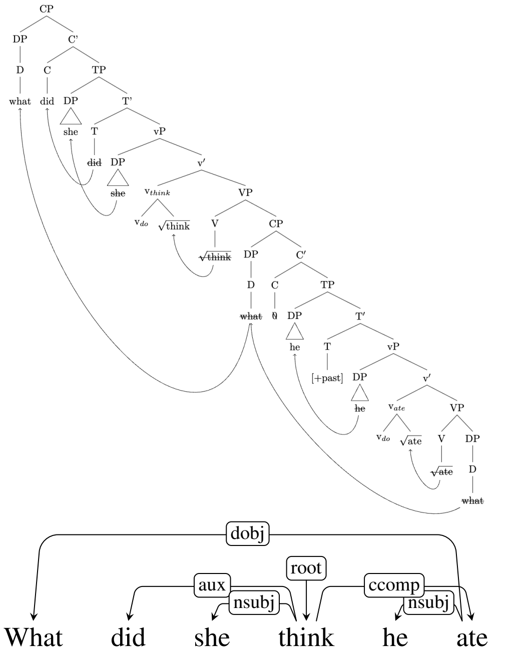
1. Pronouns: you/I/she/he/they
2. Nouns: the {teacher/student/woman/man/people}
3. Modified Nouns: the {brilliant teacher/new student/clever woman/smart man/rowdy people}
4. Possessive Noun: {the modified noun}'s friend[14]

B Minimalist Trees

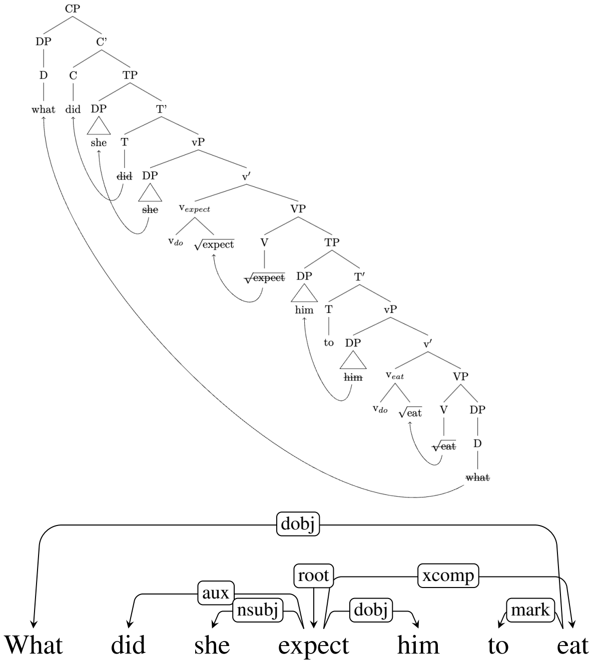
For illustrative purposes, we have utilized verb-flavors and roots from the school of Distributed Morphology. However, this is not of importance to the hierarchical distance as it is calculated from the merged result of the root and verb flavor. Other theoretical representation choices are a consequence of personal ideology, but does not impact the critical distinction that CP > TP > vP/VP. Two analyses for perception verbs are provided: one which utilizes a bare vP à la Sheehan and Cyrino (2023) (Example (9)) and one which utilizes a bare TP like Felser (1998) proposes (Example (8)). Our work favored the bare vP analysis—and furthermore found support for such an analysis—but a discussion on the two approaches can be found in Appendix C.

While not included, DoubTP and DoubCP trees contained hierarchical distance of approximately 18 and 22 and follow the same tree diagramming as illustrated in Examples (6)-(9).

(6) CP Complement *("what" and "eat" constit dist* ≈ 15*; dep dist = 1)*

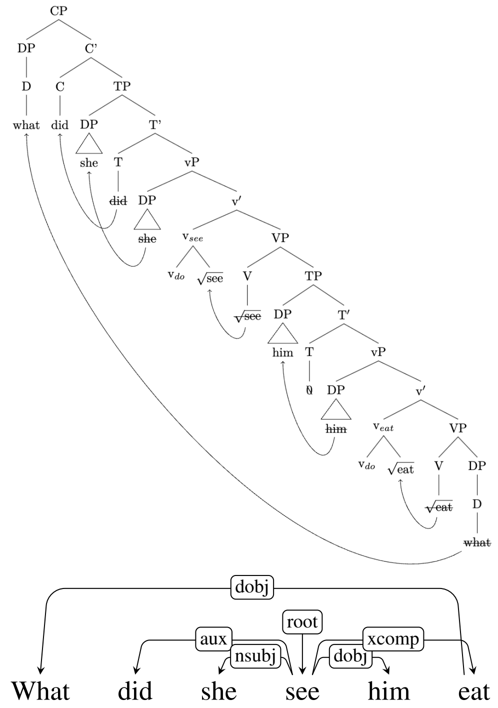


(7) ECM TP Complement *("what" and "eat" constit dist* ≈ 13*; dep dist = 1)*

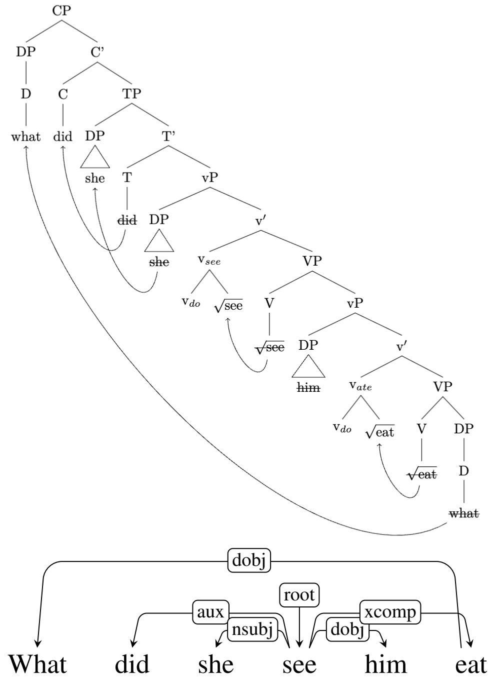


[14] When necessary for BareVP and SingCP, an adverb was inserted before the mostly deeply embedded verb.

(8) Bare TP Complement *("what" and "eat" constit dist* ≈ 13*; dep dist = 1)*



(9) Bare vP Complement *("what" and "eat" constit dist* ≈ 11*; dep dist = 1)*



C Further Analyses

Examining only TP and BareVP's difference from CP complements may not fully suggest that constituency structures are captured by pretrained language models. If we look only at vP/TP versus CP, it is possible that it is simply that BERT and the dependency probe are sensitive to finiteness, with CP being a finite phrase and vP/TP being non-finite.

Even under this possible interpretation, the implications for Dependency Grammar would be significant. Various theories of Dependency Grammar have postulated different treatments of the matter of finiteness; Lexicase (Starosta, 1988) and Word Grammar (Hudson, 1984) incorporate case relations in order constrain case assignment, which helps to assist in determining finiteness in English since finite verbs are generally conceived of as assigning nominative case in addition to incorporating features that help to distinguish the two structures (Starosta, 1997). However, the distinction between the two is not well discussed, and there exists no discussion that would explain why a verb embedded under a finite CP complement would be represented as being further away from a moved *wh*-constituent compared to a nonfinite TP or vP complement in the Chomskyan syntax. That CP complements show a further distance from their non-finite counterparts is already well captured and explained in constituency-based theories; that the dependency probe is sensitive to such distinctions in their representation is worth pursuing in the Dependency Grammar framework in order to explain this new data.

Additionally, the complements of perception verbs have been debated amongst constituent linguists (see Felser (1998) for bare infinitival TP argument and see Sheehan and Cyrino (2023) for bare vP argument analysis). Looking only at Figure 3, the distances for perception verb condition and singular ECM appear similar. However, analyses reveal statistically significant behavior in which ECMs showed significantly longer distances. Given that neither are finite, it becomes difficult to posit that the difference is due to some non-finite quality. This leads us to suspect that such differences are perhaps linked to a constituency-based analysis in which perception verbs take a complement whose size is smaller than that of the well-established TP phrase in ECM constructions, which lends support for the analysis in Sheehan and Cyrino (2023).

D Extra Figures and Results

The probe model frequently did not establish a dependency relationship between the direct object (the *wh*-question word) and the most deeply embedded verb, achieving undirected unlabeled accuracy scores far lower than those reported in Hewitt and Manning (2019), which ranged from 79.8%-82.5%, depending on the model probed. This low accuracy is likely due to various elements, such as the linear distance being a negative factor (accuracy worsens with increased linear distance, which is a well-known feature, or bug rather, of LLMs and their bottle-neck struggle to handle long-range dependencies) as well as questions being poorly represented in probe's training data and therefore more prone to inaccurate parsing. The probe's performance on the various conditions can be seen in Table 2.

In general, DoubTP achieves consistently low performance, even at the first initial and simplest iteration (0.218 for a sentence such as "What did you expect her to require him to eat?"), which is perhaps unsurprising as this construction is rather rare in natural data and is unattested in the probe's training data from the Penn Treebank (Marcus et al., 1993), which utilizes newspaper articles, which is inherently less likely to include questions, particularly those that are extracted out of doubly-embedded clauses. Similar performance appears—likely for similar reasons—with the doubly-embedded CP (DoubCP) which likewise performs poorly even at the simplest form (0.167 for a sentence like "What do you believe she thought he ate?"). Improving performance on these structures is worth further research.

LinDist		**BarevP**	**SingTP**	**SingCP**	**DoubTP**	**DoubCP**
4	Total	416		520		
	Corr.	144		285		
	Acc.	0.3462		0.5481		
5	Total	832	312	520		
	Corr.	185	150	137		
	Acc.	0.2224	0.4808	0.2635		
6	Total	728	312	624		840
	Corr.	148	104	199		140
	Acc.	0.2033	0.3333	0.3189		0.1667
7	Total	728	312	624		858
	Corr.	128	95	195		114
	Acc.	0.1758	0.3045	0.3125		0.1329
8	Total	728	624	624	702	858
	Corr.	58	261	88	153	114
	Acc.	0.0797	0.4183	0.1410	0.2179	0.1329
9	Total	728	624	624	702	858
	Corr.	178	160	180	68	141
	Acc.	0.2445	0.2564	0.2885	0.0969	0.1643
10	Total	728	624	624	702	858
	Corr.	174	134	172	47	82
	Acc.	0.2390	0.2147	0.2756	0.0670	0.0956

Table 2: The total number of sentences generated (Total) per condition per linear distance for the structural probe experiment. The number of sentences that correctly established a dependency between the *wh*-question word and the deepest embedded verb is also listed (Corr). Additional sentences were added as needed in order to achieve at least approximately 50 sentences. The percentage of sentences that correctly established the proper dependency relationship is also recorded (Acc.).

Discourse Sensitivity in Attraction Effects: The Interplay Between Language Model Size and Training Data

Sanghee J. Kim
University of Chicago
sangheekim@uchicago.edu

Forrest Davis
Colgate University
fdavis@colgate.edu

Abstract

While work on the linguistic ability of language models (LMs) is driven by a variety of aims, one dominant motivation is using LMs to determine what linguistic knowledge can be learned from unstructured text. The current work aims to evaluate LMs on discourse sensitivity—the capability to distinguish between content that is more relevant and important to the discourse and that which is less so. We ground our evaluation of LMs by leveraging an existing psycholinguistics study on the *number agreement attraction effect*, one of the well-studied measures of human language comprehension. Based on human empirical findings on the modulation of the attraction effect by discourse, we establish three tests that LMs should pass if they demonstrate discourse sensitivity. A total of 25 models were evaluated that vary in (i) model size (small or large) and (ii) training type (dialogue-based, plain, and instruction-based). The models showed systematicity in discourse sensitivity, though in ways dissimilar to humans, either by over-relying on structural cues or overusing discourse cues. Notably, models that patterned most similarly to human performance were predominantly smaller and those trained on dialogue-targeted data. We discuss the implications of these findings and insights into human language processing.

1 Introduction

A growing body of work has investigated the linguistic capabilities of language models (LMs), tackling aspects of syntax, semantics, and pragmatics (for a survey, see Chang and Bergen, 2024). While work on the linguistic ability of LMs is driven by a variety of aims, one dominant motivation is using LMs to determine what linguistic knowledge can be learned from unstructured text (Linzen and Baroni, 2021). Some work has claimed LMs obtain abstract linguistic knowledge, resolving complex syntactic (e.g., Wilcox et al., 2024) and anaphoric dependencies (e.g., Hu et al., 2020), and exhibiting signs of pragmatic skills (e.g., Hu et al., 2023), though there is nuance in what can be inferred from these types of results (for a case study in the limitations of inferring full grammatical knowledge from overlap in behavior, see Lan et al., 2024).

Much of the work on linguistic evaluations of LMs focuses on linguistic phenomena treated in isolation. For example, linguistic knowledge benchmarks like BLiMP (Warstadt et al., 2020) and SyntaxGym (Gauthier et al., 2020) explore linguistic phenomena separately (e.g., subject-verb agreement, argument structure) rather than the interaction of multiple processes (e.g., interactions between argument structure and agreement; for discussion see Davis (2022b)). The current study aims to expand on this body of work by investigating the interaction of discourse structure with syntactic dependencies. We ask whether exposure to a massive amount of text and differing forms of training (e.g., instruction finetuning) yields “knowledge” of discourse.

Building on a large body of work investigating subject-verb agreement in language models (Linzen et al., 2016; Arehalli and Linzen, 2020; Warstadt et al., 2020; Yedetore and Kim, 2024, a.o.), we focus on structures like the following:

(1) The waitress *who sat near the girls* was unhappy.

In (1), the agreement between the main verb (*was*) and the subject (*The waitress*) can be made difficult because of an interfering noun, *girls*, which, if misidentified as the subject, would yield a different agreement pattern (e.g., *were*). This influence of interfering nouns, when the subject-verb agreement needs to be resolved, leads to an *interference effect* and has been widely used in both human studies (e.g., Wagers et al., 2009) and evaluations of language models (e.g., Arehalli and Linzen, 2020).

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 287-299.
Eugene, Oregon, July 18-20, 2025

In our study, we manipulate the discourse status of the relative clause containing the interfering noun. In (1), the relative clause (*who sat near the girls*) is a restrictive relative clause. Restrictive relative clauses conventionally convey essential information to the discourse (as they function as selecting a specific referent). By adding commas surrounding the relative clause (i.e., *The waitress, who sat near the girls, was unhappy*), we can signal an appositive relative clause, which conveys side-commentary information and are not part of the main assertion (Potts, 2005; AnderBois et al., 2015; Syrett and Koev, 2015; Koev, 2022; cf. Potts, 2012). We make use of this contrast in discourse status between the two structures to examine the interaction of discourse structure and syntactic dependencies, specifically cases where human processing of subject-verb agreement is modulated by discourse status.

As argued for in Suijkerbuijk et al. (2024), we ground our evaluation of language models via comparison to an existing psycholinguistic study demonstrating human *discourse sensitivity* (Kim and Xiang, 2024). Drawing on the same materials, we established three tests that LMs should pass to exhibit human-like behavior. Concretely, we investigated 25 models, including plain (base) and instruction-tuned models, and models trained on dialogue and conversational goal-oriented datasets, and those that vary in model size (small or large).

To preview the findings, the results suggest that (i) models trained on datasets with dialogue and goal-oriented conversations outperform other models, (ii) larger models do not yield human-like discourse sensitivity, and (iii) instruction-based training does not necessarily benefit models compared to base training. Taking these findings, we suggest that the *qualitative* nature of training data (e.g., genre and the specific types of constructions) is critical in the success of discourse sensitivity. We conclude by discussing insights into human language processing from evaluating language models and why instruction-tuned models underperform compared to base models, despite their seemingly advantageous training.

2 Background

2.1 Discourse structure: the division of more and less important information

Discourse can be defined in multiple different ways. It can be illustrated as a coherence relation (Hobbs, 1985; Kehler, 2002), the conversational moves for a successful discourse (Lewis, 1979; Farkas and Bruce, 2010), a hierarchically structured representation of discourse units (Polanyi, 1988; Asher and Lascarides, 2003; Jasinskaja, 2016), or a set of organized question and answer pairs to the conversational topic (Roberts, 2012), to name a few.

Regardless of the approaches to analyzing discourse, however, a shared notion of discourse is that certain parts of discourse are more important than others—some components of discourse are more relevant to the discourse topic, and others are less so. The examples in (2) demonstrate this contrast, realized at a sentence level:

(2) a. The waitress *who sat near the girl* was unhappy. [RRC]
b. The waitress, *who sat near the girl*, was unhappy. [ARC]

The same content, *that the waitress sat near the girl* is primary discourse information in (2a), essential to specify the very waitress that is being discussed, whereas it is secondary information in (2b), adding side-commentary details to the discourse. This division is expressed with the contrast of restrictive (RRC) (2a) and appositive relative clauses (ARC) (2b). Throughout the paper, we use these two structures to distinguish between different types of discourse status, serving as stand-ins for discourse structure at the sentence level.

2.2 Human sensitivity to discourse

Theoretical and experimental studies have shown that humans are highly sensitive to distinctions in information status (Potts, 2005; Syrett and Koev, 2015). In ongoing discourse, content that is part of the main discourse structure is judged to be a more natural continuation than content belonging to a non-main or subordinate discourse structure, such as information in an appositive relative clause (Syrett and Koev, 2015; Göbel, 2019). Discourse salience, topichood, and coherence have also been shown to affect real-time language comprehension and production. Entities that are salient in discourse are easier to recall and retrieved (e.g., Birch and Garnsey, 1995; Sturt et al., 2004), those in topical or focused sentential positions are more likely to be selected as antecedents of pronouns (e.g., Arnold, 1998; Kaiser, 2011; Rohde and Kehler, 2014; Colonna et al., 2012), and discourse topic (or Question under Discussion) modulates the ease of comprehension (e.g., Clifton and Frazier, 2012;

Kehler and Rohde, 2017; Clifton and Frazier, 2018) and the resolution of syntactic ambiguity (e.g., Kehler, 2015). Additionally, these distinctions have immediate effects on processing, with studies demonstrating their active use in real-time language comprehension. For instance, when linguistic materials known to lead to processing difficulty (e.g., long embedded relative clauses) are part of less important discourse, they result in reduced processing difficulty (Dillon et al., 2014, 2017; Kroll and Wagers, 2019; Duff et al., 2023).

2.3 Language model sensitivity to discourse

While the linguistic evaluations of language models have been dominated by syntactic contrasts (for a survey, see Chang and Bergen, 2024), there has been a growing body focusing on discourse knowledge. This includes work on the interaction of discourse structure and pronouns (e.g., Davis and van Schijndel, 2020), discourse structure and at-issueness (e.g., Kim et al., 2022), implicatures and presuppositions (e.g., Jeretic et al., 2020), and discourse connectives (e.g., Cong et al., 2023; Pandia et al., 2021). Broadly, pre-trained language models appear to capture some contextual effects. However, there are still notable differences between model and human behavior, suggesting differences in their processing of discourse. More recently, the impact of instruction-tuning on the linguistic knowledge of models has been investigated, with some results showing that such fine-tuning results in models with a worse fit to human behavioral measures (Kuribayashi et al., 2024). Moreover, the exact fine-tuning strategy directly impacts the ability of models on discourse tasks, with some strategies yielding models with better pragmatic abilities (Ruis et al., 2024). These results suggest that, while instruction-tuning was proposed to align models with human discourse preferences, it may not always align with the linguistic behavior of humans. The present study finds additional support for this misalignment.

3 Metrics

3.1 Interference effect

To evaluate model performance on its discourse sensitivity, we compare the *interference effect*, a common way to show the cognitive process that underlies human language comprehension (Van Dyke and Lewis, 2003; Lewis and Vasishth, 2005). For example, the interference effect is observed in the different degrees of acceptance of the two sentences in (3), even when both are ungrammatical. Studies have found that (3b) is considered more acceptable than (3a), and reading times at the verb (*were*) are commonly found to be faster in (3b) compared to (3a) (Wagers et al., 2009; Parker and An, 2018, a.o). Such a difference between the two ungrammatical sentences derives from the interfering linguistic unit, *the girl(s)*, where the plural (number) feature of *the girls* matches the feature of the verb (*were*)—leading to a *number agreement attraction effect*.

(3) a. *<u>The waitress</u> who sat near *the girl* <u>were</u> unhappy.
b. *<u>The waitress</u> who sat near *the girls* <u>were</u> unhappy.

Empirical findings suggest that this effect primarily occurs in ungrammatical sentences—when the subject and verb do not match (e.g., *the waitress... were* instead of *was*) (e.g., Wagers et al., 2009; cf. Jäger et al., 2017)—commonly referred to as the *standard number agreement attraction effect*.

Interference effect in human reading times In this study, we use the number agreement attraction effect as a signal of human processing, typically measured by the difference in reading time (RT) between the singular (3a) and plural (3b) conditions, subtracting the singular from the plural condition:

$$\text{Interference effect} = RT_{\text{plural}} - RT_{\text{singular}} \qquad \text{(Eq. 1)}$$

Interference effect in models Following previous work (e.g., van Schijndel and Linzen, 2021), we used language model surprisal in correspondence to human reading time. Surprisal (Hale, 2001) is defined as (Eq. 2), calculated at the critical verb position given prior context left of the verb. The surprisal was calculated from the logits of the model.[1]

$$\text{Surprisal} = -\log P(\text{verb} \mid \text{left context}) \qquad \text{(Eq. 2)}$$

In examining the number agreement attraction effect, we evaluated the difference in surprisal at the verb between the context with a plural distractor (i.e., plural context) and a singular distractor (i.e., singular context), as shown in (Eq. 3). For example, the difference in surprisal for the verb *were* when the left context was *The waitress who sat near the*

[1] For GODEL-based models, we calculated the surprisal using the decoder of the encoder-decoder model.

girls and when the left context was *The waitress who sat near the girl* was calculated.

$$\begin{aligned}\Delta\text{Surprisal}(\text{verb}) = & \\ & -\log P(\text{verb} \mid \text{plural context}) \\ & -(-\log P(\text{verb} \mid \text{sigular context}))\end{aligned} \quad \text{(Eq. 3)}$$

For each model, the presence or absence of the effect was determined by comparing the bootstrapped 95% confidence interval (CI) of the average interference effect as in (Eq. 4).[2]

$$\text{Average Interference Effect} = \frac{1}{N}\sum_{i=1}^{N} \Delta\text{Surprisal}_i(\text{verb}), \quad \text{(Eq. 4)}$$

where N is the number of samples.

The absence of an interference effect was determined by whether the CI overlapped with zero (i.e., there was no difference between the plural and singular conditions).

3.2 Evaluation of discourse sensitivity

The attraction effect has been reported to be robustly found when the distractor noun is linearly close to the verb (*the girls ... were* as in (4a)) and even when it is distant (*the musicians ... praise* as in (5a)) (e.g., Wagers et al., 2009). Studies have further found that the attraction effect, however, can be modulated by the discourse status of the distractor noun, where in one case, the standard attraction effect disappears (4b) (Ng and Husband, 2017; McInnerney and Atkinson, 2020; Duff et al., 2023; Kim and Xiang, 2024) but it sustains in the other (5b) (Kim and Xiang, 2024).

(4) a. *The waitress who sat near *the girls* were unhappy.
b. *The waitress, who sat near *the girls*, were unhappy.

(5) a. **The musicians* who the reviewer praise highly will win a Grammy.
b. **The musicians*, who the reviewer praise highly, will win a Grammy.

When the distractor (e.g., *the girls*) is part of secondary information as in (4b), it does not interfere when the subject-verb dependency needs to be resolved, and hence the number agreement attraction effect is absent. On the contrary, when the distractor (e.g., *the musicians*) is related to the discourse topic (or Question under Discussion as in Roberts (2012)) at retrieval (e.g., *praise*) in (5b)), the distractor interferes and leads to a number agreement attraction effect (Kim and Xiang, 2024). This modulation of the interference effect due to the discourse status of the distractor noun will be used as a signal for *discourse sensitivity*.

Discourse sensitivity in human reading times The key aspects of discourse sensitivity in humans in interference effects are summarized in Table 1. First, in both constructions (Experiments 1 and 2), a standard attraction effect is found with the baseline RRC condition. This is identified by significant reading differences between the singular and plural distractor conditions in the ungrammatical condition but not in the grammatical condition (Eq. 5).

$$\text{Standard number agreement attraction effect} = \begin{cases} RT_{\text{plural}} - RT_{\text{singular}} < 0 & \text{if ungrammatical,} \\ RT_{\text{plural}} - RT_{\text{singular}} \simeq 0 & \text{if grammatical.} \end{cases} \quad \text{(Eq. 5)}$$

Secondly, the standard attraction effect should be present in structures as in (4b) (Experiment 1) but absent in structures as in (5b) (Experiment 2).

Discourse sensitivity in models Using the above-mentioned human reading time results identifying discourse sensitivity as a baseline (Kim and Xiang, 2024), we evaluate model outputs based on the three following tests:[3]

- **Discourse Attraction**. In Experiment 1, the standard attraction effect is exhibited in the RRC structure (4a) but not in the ARC structure (4b). For RRCs, the average difference in surprisal between the plural distractor and the singular distractor should be negative, indicating that plural distractors lower the surprisal of plural verbs. ARCs should be significantly different from RRCs with RRCs exhibiting a larger interference effect (i.e., more negative).
- **Standard Attraction**. In Experiment 2, the standard effect is exhibited in both the RRC (5b) and ARC (5b) structures. The average difference in surprisal should be negative, indicating that plural distractors lower the surprisal of plural verbs. More specifically,

[2] Bootstrapping was done with 1000 samples and resampling.

[3] For code and data: https://github.com/sangheek16/discourse-sensitivity-attraction-effect.git.

Exp.	Clause	Grammaticality	Input (subject-verb is **bold-faced**; distractor is underlined)	Effect
1	RRC	Grammatical	**The waitress** who sat near the girl(s) **was** unhappy.	✗
1	RRC	Ungrammatical	**The waitress** who sat near the girl(s) **were** unhappy.	✓
1	ARC	Grammatical	**The waitress**, who sat near the girl(s), **was** unhappy.	✗
1	ARC	Ungrammatical	**The waitress**, who sat near the girl(s), **were** unhappy.	✗
2	RRC	Grammatical	The musician(s) who **the reviewer praises** will win a Grammy.	✗
2	RRC	Ungrammatical	The musician(s) who **the reviewer praise** will win a Grammy.	✓
2	ARC	Grammatical	The musician(s), who **the reviewer praises**, will win a Grammy.	✗
2	ARC	Ungrammatical	The musician(s), who **the reviewer praise**, will win a Grammy.	✓

Table 1: Human baseline: presence (✓) vs. absence (✗) of interference effect (Kim and Xiang, 2024).

we divide this test into two subcases. With the stronger version of this test (**Standard Attraction-Strong**), the magnitude of the interference effect between RRC and ARC should be comparable. In the weaker version (**Standard Attraction-Weak**), the size of the interference effect does not matter as long as both exhibit an attraction effect.

- **Grammatical Asymmetry**. As a signal for a standard number agreement attraction effect, there should be no interference effect (i.e., no difference based on whether the distractor is singular or plural) in all grammatical conditions, regardless of clause type and experiment.[4]

4 Model selection

We tested 25 models for evaluation, either base-trained or involving instruction-based tuning, and varying in size and type of training data. Models are categorized below and summarized in Table 2.

Categorization 1: Based on the number of parameters First, we compared models that vary in their number of parameters. Specifically, we examined whether larger models yield performance similar to that of humans. Given the current emphasis in the field of scale, we might straightforwardly predict that bigger models are more likely to pass the tests. However, some empirical results suggest that scale does not necessarily mean better prediction of human behavior (e.g., Oh and Schuler, 2023; Oh et al., 2024; Shain et al., 2024; Wilcox et al., 2024). As Oh et al. suggest, LLMs make good predictions on words with low frequency, which in turn is not what is expected in human data. If the same type of counter-advantage of large models applies to examining discourse sensitivity, then we could see better performance with smaller models.

Categorization 2: Based on the genre of data Second, we also examine whether the genre of data would affect the performance of LMs in discourse sensitivity. Earlier work has shown that models outperform others in dialogue and discourse settings when trained on data with conversation and naturalistic data (Wolf et al., 2019; Bao et al., 2020; Henderson et al., 2020; Wu et al., 2020; Zhang et al., 2020; Gu et al., 2021; Thoppilan et al., 2022). We acknowledge that the comparison of the genre of data between models is not totally straightforward, especially given the lack of accessibility to LLMs' training data. For example, the training data used for some LLMs may include the data used for the "dialogue-based models." However, we believe a useful comparison can still be sustained. If the dialogue-based models outperform the models trained on a variety of genres, then we take this as evidence that training data primarily composed of discourse-goal and dialogue-oriented data is of better quality, for alignment with human linguistic behavior, than a larger composition of varied genres and styles.

Categorization 3: Based on tuning/training type Finally, we also examine whether instruction models outperform base models in discourse sensitivity.

[4] We acknowledge prior findings showing that the grammatical asymmetry in attraction effects—typically observed in ungrammatical conditions—can be influenced by task factors such as response bias and answer ratios (e.g., Hammerly et al., 2019; Laurinavichyute and von der Malsburg, 2024). These studies found that the asymmetry is masked when the response bias is neutralized. However, in the study by Kim and Xiang (2024), which provides the human reading time data used for model evaluation in the current work, the task was explicitly designed to neutralize response bias. Therefore, while we recognize this as a general concern in the literature, we suspect it is less likely to impact the interpretation of our current results.

Given that instruction-tuned models are arguably better at capturing the user's (or the interlocutor's) needs and goals (see Zhang et al. (2023) for an overview), we speculate that models could benefit from such training methods to achieve better performance in discourse sensitivity, similar to understanding discourse goals. They could demonstrate patterns that align well with human expectations in discourse and dialogue settings. Yet, there is only little work on investigating how well instruction-based models align with human behavior. While instruction tuning can result in greater alignment at the high-level representation (e.g., between the LLM internal representation and human neural activity, see Aw et al., 2024), findings also suggest that at the behavioral level, there is no model-human alignment such as in human reading times or judgment tasks (Zhang et al., 2023; Kauf et al., 2024; Aw et al., 2024). Given that discourse sensitivity in the current work is measured through surprisal and is compared against human reading time data, it is possible that instruction-based models would not outperform the base models.

5 Results

Table 2 shows the list of models we evaluated and the results.

5.1 By each test

Discourse Attraction. With only one exception of DialoGPT-small, all dialogue-based models passed this test. While GPT-Neo-125M, GPT-Neo-2.7B, and Mistral-7B-v0.3 passed Discourse Attraction, the remaining models did not, showing no systematic correlation with training type or size.

Standard Attraction. All but three models passed Standard Attraction-Weak. The models that did not pass this test are all small dialogue-based models: DialoGPT-large, GODEL-base, and GODEL-large. When a stronger version (Standard Attraction-Strong) was applied, four additional models failed to pass: GPT-J-6B, Mistral-7B-v0.1, Mistral-7B-v0.3, and Mistral-7B-Instruct-v0.3.

Grammatical Asymmetry. None of the models passed Grammatical Asymmetry. All models exhibited an interference effect in the grammatical condition of at least one of the clause types in at least one of the experiments.

5.2 By combined tests

To better understand the results, we analyze them by each of the four combinations that can be found in passing Discourse Attraction and Standard Attraction. The models' failure to pass Grammatical Asymmetry is discussed in Section 6.

Discourse Attraction:✓, Standard Attraction:✓. This is a case where models were most sensitive to discourse division. Passing both of these tests signals a division of primary versus secondary information driven by the syntactic difference between RRC and ARC structures—as in Discourse Attraction—while not simply making distinctions between RRC and ARC structures based on their syntactic form—as in Standard Attraction. *Models*: DialoGPT-medium, GPT2-small, GPT-Neo-125M, and GPT-Neo-2.7B. The models that passed both of these tests (Discourse Attraction and Standard Attraction-Strong) were all small GPT-based models.

Discourse Attraction:✓, Standard Attraction:✗. This is a case where models were sensitive to the division between RRC and ARC and were applying the same division to resolving the linguistic dependency in Experiment 2. However, as we have seen in human performance, it is not simply the syntactic division between RRC and ARC to pass Standard Attraction; the interference effect with the ARC condition that was absent in Experiment 1 should be present in Experiment 2. The models under this category did not exhibit that contrast, suggesting that while they have grasped the syntactic division, the nuanced discourse division was not captured. *Models*: DialoGPT-large, GODEL-base, GODEL-large. These models were exclusively small, dialogue-based models.

Discourse Attraction:✗, Standard Attraction:✓. This is a case where models exhibited an interference effect in both experiments in both clauses. While all models showed the baseline interference effect in the RRC condition, the failure to pass Discourse Attraction was driven by the presence of the interference effect in the ARC condition. The results can be interpreted in that while the models showed an interference effect, they lacked discourse or syntactic division. *Models*: DialoGPT-small, GPT2-medium, GPT2-large, GPT2-XL, GPT-Neo-1.3B, GPT-J-6B, Llama-2-7B, Llama-2-13B, Llama-3-8B, Llama-3.1-8B, Mistral-7B-v0.1, Llama-2-7B-Chat, Llama-2-13B-Chat, Llama-3-

Training Type	Size	Model	Size	Discourse	Standard-weak	Standard-strong	Grammatical
Dialogue	Small	DialoGPT-small	117M	✗	✓	✓	✗
		DialoGPT-medium	345M	✓	✓	✓	✗
		DialoGPT-large	762M	✓	✗	✗	✗
		GODEL-base	220M	✓	✗	✗	✗
		GODEL-large	770M	✓	✗	✗	✗
Plain		GPT2-small	124M	✓	✓	✓	✗
		GPT2-medium	355M	✗	✓	✓	✗
		GPT2-large	774M	✗	✓	✓	✗
		GPT2-XL	1.5B	✗	✓	✓	✗
		GPT-Neo-125M	125M	✓	✓	✓	✗
		GPT-Neo-1.3B	1.3B	✗	✓	✓	✗
		GPT-Neo-2.7B	2.7B	✓	✓	✓	✗
		GPT-J-6B	6B	✗	✓	✗	✗
	Large	Llama-2-7B	7B	✗	✓	✓	✗
		Llama-2-13B	13B	✗	✓	✓	✗
		Llama-3-8B	8B	✗	✓	✓	✗
		Llama-3.1-8B	8B	✗	✓	✓	✗
		Mistral-7B-v0.1	7B	✗	✓	✗	✗
		Mistral-7B-v0.3	7B	✓	✓	✗	✗
Instruction		Llama-2-7B-Chat	7B	✗	✓	✓	✗
		Llama-2-13B-Chat	13B	✗	✓	✓	✗
		Llama-3-8B-Instruct	8B	✗	✓	✓	✗
		Llama-3.1-8B-Instruct	8B	✗	✓	✓	✗
		Mistral-7B-Instruct-v0.1	7B	✗	✓	✓	✗
		Mistral-7B-Instruct-v0.3	7B	✗	✓	✗	✗

Table 2: Model comparison: passed (✓) vs. failed (✗) the test.

8B-Instruct, Llama-3.1-8B-Instruct, Mistral-7B-Instruct-v0.1, Mistral-7B-Instruct-v0.3. These include all the instruction-based models, most of the large models, and most of the small plain models.

Discourse Attraction:✗, Standard Attraction:✗. This would be the case where models demonstrated no interference effect. No model exhibited this behavior, which confirms that they were influenced by the presence of a distractor noun in at least some conditions. All models demonstrated the baseline effect of interference in the ungrammatical RRC condition. *Models*: None.

6 Discussion

No models passed all three tests. However, all models were influenced by distractors, facilitating the use of interference effects to test whether discourse structure influenced model's predictions. While the models did not pass all three tests, they showed systematicity in their performance on Discourse Attraction and Standard Attraction. In one case ({Discourse Attraction: ✗, Standard Attraction: ✓}), the presence of a distractor led to an interference effect, but this effect was not modulated by discourse division. In the other case ({Discourse Attraction: ✓, Standard Attraction: ✗}), the models were guided by the discourse, or the syntactic division of RRC and ARC, but they were overapplying this division. Four of the tested models performed in the most principled way, where they passed Discourse Attraction and Standard Attraction: DialoGPT-medium, GPT2-small, GPT-Neo-125M, and GPT-Neo-2.7B. In the following, we elaborate on the less principled models.

Lack of discourse division (Discourse: ✗, Standard: ✓). This is a systematic pattern where the models show the standard number agreement attraction effect without showing sensitivity to discourse division. Models showed the attraction effect in all constructions in (4)–(5), indicating that the different discourse status of the distractor in (5b) was not considered. This pattern was prevalent in most of the models, except for the small dialogue-based models. This is in line with earlier studies that have shown cases where grammatically irrelevant words modulate the surprisal at the critical word (in subject-verb agreement (Ryu and Lewis, 2021; Arehalli and Linzen, 2020) as well as reflexive pronoun resolution (Ryu and Lewis, 2021; Davis, 2022a). The influence of linearly closer, but grammatically irrelevant words, remains a feature of even the larger models. That is, increases in scale and other training approaches have not made models robust to interference effects.

Heavy reliance on syntactic/discourse division (Discourse: ✓, Standard: ✗). In line with the finding discussed above, it is still the case that all models under this category have exhibited the standard number agreement attraction effect in the baseline RRC condition (as in (4a) & (5a)). Nonetheless, the effect was not present with the ARC structure in both Experiment 1 (as in (4b)) and Experiment 2 (as in (5b)), suggesting that it is possible that models heavily relied on the linguistic cue that distinguishes the main content from the subordinate content in the sentences with the ARC structure. Earlier work using a probing task showed that LMs successfully classify (with greater than 99% accuracy) the main content differently from the subordinate content (Kim et al., 2022). Hence, it is possible that the structural difference (or even simply the presence of commas) of ARCs compared to RRCs has resulted in the absence of the attraction effect.

However, there is another possibility beyond the models tracking the superficial cues or the syntactic representation: the models were (overly) applying discourse division cues. Recall that the only three models that fell under this category are DialoGPT-large, GODEL-base, and GODEL-large, all trained on dialogue-based data. We conjecture that it is not coincidental that the overapplication of the division of main versus subordinate content to attraction effect was only found in the dialogue-based models. We speculate that the specific training process has led to an effect of models exhibiting abstract signals about discourse structure, either (a) naturally following from the abstract structural representation through training, or (b) demonstrating a separate pattern that is learned in addition to the abstract structural representation.

Given the promising performance of recent instruction-based models, it is perhaps unexpected that they fall short in exhibiting the level of discourse sensitivity in humans. This discrepancy may stem from the training methods of these instruction-based models, which are optimized for extracting and producing the most relevant information efficiently and concisely. During training, they are directed to perform tasks such as summarization and a clear question and answering (Zhang et al., 2023). However, human discourse includes purposeful digressions—often for the richness of conversation—and layers of primary (main) and secondary (subordinate) information. The different conversational goal perhaps accounts for the reason why instruction-based models diverge from the discourse division that humans show.

Why didn't any of the models pass Grammatical Asymmetry? Grammatical Asymmetry examined whether models exhibit the standard number agreement attraction effect, i.e., whether the attraction effect is found only in the ungrammatical and not in the grammatical condition. One of the ways to account for the asymmetric attraction effect in humans is an error-driven process, where the interference effect is realized only when there is a mismatch between the retrieval target (i.e., subject) and the retrieval site (i.e., verb)—that is, when the sentence is ungrammatical (Wagers et al., 2009; Lago et al., 2015; Schlueter et al., 2019). However, such an error-driven process seems unlikely for the models. As we saw in the results with Grammatical Asymmetry, the presence of the distractor in the subject-verb dependency led to an attraction effect, even when the subject and the verb agreed—that is, when the sentence was grammatical, and hence there were no "errors."

The contrast between human and model performance has implications for interpreting the models, where the distractor does not have an equal status in language processing. Humans may be applying a top-down approach (by incorporating the discourse status of distractors) (e.g., Kutas et al., 2011) while incorporating bottom-up linguistic information (Momma and Phillips, 2018) (such as number information). While prediction and expectation on the verb that agrees with the subject are formed in real time in humans, models are strongly driven by a bottom-up incremental process, where the linear sequence of the incoming linguistic units is influential on the retrieval process.

7 Conclusion

The current work examined the discourse sensitivity of language models by investigating the interaction between discourse structure and syntactic dependency. Leveraging findings from human experiments on the number agreement attraction effect, we compared language model behavior to human behavior. Critically, the pattern we targeted was the presence of a standard attraction effect in Experiment 1 (Discourse Attraction), its absence in Experiment 2 (Standard Attraction), and the presence of a grammatical asymmetry (Grammatical Asymmetry). As discussed in Kim and Xiang (2024), humans show a modulated attraction effect across the two experimental setups, driven by their sensitivity to the active discourse question (akin to the Question Under Discussion).[5] None of the 25 models fully overlapped with humans: some models associated structural cues with discourse, while others overapplied discourse cues. Larger models exhibited the attraction effect in both Experiment 1 and Experiment 2, indicating insensitivity to the nuanced discourse status of distractor and target noun phrases. In contrast, smaller models trained on dialogue-based data showed the best performance—even outperforming large, instruction-based models. These smaller models exhibited a modulated attraction effect, suggesting they may have learned some abstract representation of discourse, though not fully matching human retrieval patterns, as shown by their failure in Grammatical Asymmetry. As discussed in the Discussion section, we conjecture that larger models may underperform relative to smaller models in capturing human-like patterns due to the scale of their training data. Furthermore, instruction-tuned models may lack alignment with human discourse goals and conversational dynamics given their training objective.

Future work could solidify these claims by surveying a larger variety of instruction-tuning approaches and carefully controlling the training data to tease apart the effect of data quality on model performance (as in Misra and Mahowald, 2024). Ultimately, the contrast between language processing in humans and machines highlights a disconnect in their abilities to integrate multiple sources of information. While humans combine syntactic and discourse information, and top-down and bottom-up linguistic signals, models overrely on one of these sources.

8 Limitations

The current study used a discrete categorization based on the absence or presence of the number agreement attraction effect. While this approach offers ease of interpretation, we acknowledge that it limits the ability to perform more quantitative evaluations. Future work could adopt a quantitative approach to compare the magnitude of the attraction effect in human reading times and surprisal across experiments. Furthermore, we focused on one specific case study to investigate models' discourse sensitivity, rather than a suite of tests. As such, the conclusions drawn from the current findings may be limited to this particular form of discourse sensitivity. The authors are developing broader tests to evaluate discourse sensitivity beyond the modulated attraction effect to assess the generalizability of the current findings.

Acknowledgments

We would like to thank Marten van Schijndel and the anonymous reviewers for their constructive comments and suggestions.

References

Scott AnderBois, Adrian Brasoveanu, and Robert Henderson. 2015. At-issue proposals and appositive impositions in discourse. *Journal of Semantics*, 32(1):93–138.

Suhas Arehalli and Tal Linzen. 2020. Neural language models capture some, but not all, agreement attraction effects. In *Proceedings of the 42nd Annual Conference of the Cognitive Science Society*, pages 370–376.

Jennifer E. Arnold. 1998. *Reference form and discourse patterns*. Ph.D. thesis, Stanford University, Stanford, CA.

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

Khai Loong Aw, Syrielle Montariol, Badr AlKhamissi, Martin Schrimpf, and Antoine Bosselut. 2024. Instruction-tuning aligns LLMs to the human brain. In *First Conference on Language Modeling*.

Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2020. PLATO: Pre-trained dialogue generation model with discrete latent variable. In *Proceedings of the 58th Annual Meeting of the Association*

[5] See Kim and Xiang (2024) for a detailed explanation of how the discourse question modulates retrieval processes that leads to the observed attraction effect.

for Computational Linguistics, pages 85–96, Online. Association for Computational Linguistics.

Stacy L Birch and Susan M Garnsey. 1995. The effect of focus on memory for words in sentences. *Journal of Memory and Language*, 34(2):232–267.

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

Charles Jr. Clifton and Lyn Frazier. 2012. Discourse integration guided by the 'Question under Discussion'. *Cognitive Psychology*, 65(2):352–379.

Charles Jr. Clifton and Lyn Frazier. 2018. Context effects in discourse: The question under discussion. *Discourse Processes*, 55(2):105–112.

Saveria Colonna, Sarah Schimke, and Barbara Hemforth. 2012. Information structure effects on anaphora resolution in German and French: A crosslinguistic study of pronoun resolution. *Linguistics*, 50(5):991–1013.

Yan Cong, Emmanuele Chersoni, Yu-Yin Hsu, and Philippe Blache. 2023. Investigating the effect of discourse connectives on transformer surprisal: Language models understand connectives, Even so they are surprised. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 222–232, Singapore. Association for Computational Linguistics.

Forrest Davis. 2022a. Incremental processing of Principle B: Mismatches between neural models and humans. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 144–156.

Forrest Davis. 2022b. *On the Limitations of Data: Mismatches between Neural Models of Language and Humans*. Ph.D. thesis, Cornell University.

Forrest Davis and Marten van Schijndel. 2020. Discourse structure interacts with reference but not syntax in neural language models. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 396–407, Online. Association for Computational Linguistics.

Brian Dillon, Charles Clifton Jr., and Lyn Frazier. 2014. Pushed aside: Parentheticals, memory and processing. *Language, Cognition and Neuroscience*, 29(4):483–498.

Brian Dillon, Charles Clifton Jr., Shayne Sloggett, and Lyn Frazier. 2017. Appositives and their aftermath: Interference depends on at-issue vs. not-at-issue status. *Journal of Memory and Language*, 96:93–109.

John Duff, Pranav Anand, Adrian Brasoveanu, and Amanda Rysling. 2023. Pragmatic representations and online comprehension: Lessons from direct discourse and causal adjuncts. *Glossa Psycholinguistics*, 2(1):1–52.

Donka F Farkas and Kim B Bruce. 2010. On reacting to assertions and polar questions. *Journal of Semantics*, 27(1):81–118.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76, Online. Association for Computational Linguistics.

Alexander Göbel. 2019. Final appositives at the right frontier: An experimental investigation of anaphoric potential. In *Proceedings of Sinn und Bedeutung 23*, pages 451–467. Universitat Autònoma de Barcelona, Bellaterra (Cerdanyola del Vallès).

Xiaodong Gu, Kang Min Yoo, and Jung-Woo Ha. 2021. DialogBERT: Discourse-aware response generation via learning to recover and rank utterances. *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*, 35(14):12911–12919.

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*.

Christopher Hammerly, Adrian Staub, and Brian Dillon. 2019. The grammaticality asymmetry in agreement attraction reflects response bias: Experimental and modeling evidence. *Cognitive Psychology*, 110:70–104.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2020. ConveRT: Efficient and accurate conversational representations from transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2161–2174, Online. Association for Computational Linguistics.

Jerry R. Hobbs. 1985. *On the coherence and structure of discourse*. Stanford, CA: CSLI Technical Report 85-37.

Jennifer Hu, Sherry Yong Chen, and Roger Levy. 2020. A closer look at the performance of neural language models on reflexive anaphor licensing. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 323–333, New York, New York. Association for Computational Linguistics.

Jennifer Hu, Sammy Floyd, Olessia Jouravlev, Evelina Fedorenko, and Edward Gibson. 2023. A fine-grained comparison of pragmatic language understanding in humans and language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4194–4213, Toronto, Canada. Association for Computational Linguistics.

Lena A Jäger, Felix Engelmann, and Shravan Vasishth. 2017. Similarity-based interference in sentence comprehension: Literature review and Bayesian meta-analysis. *Journal of Memory and Language*, 94:316–339.

Katja Jasinskaja. 2016. Not at issue any more. Unpublished manuscript, University of Cologne.

Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. 2020. Are natural language inference models IMPPRESsive? Learning IMPlicature and PRESupposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8690–8705, Online. Association for Computational Linguistics.

Elsi Kaiser. 2011. Focusing on pronouns: Consequences of subjecthood, pronominalisation, and contrastive focus. *Language and Cognitive Processes*, 26(10):1625–1666.

Carina Kauf, Emmanuele Chersoni, Alessandro Lenci, Evelina Fedorenko, and Anna A Ivanova. 2024. Comparing plausibility estimates in base and instruction-tuned large language models. *arXiv preprint arXiv:2403.14859*.

Andrew Kehler. 2002. *Coherence, reference, and the theory of grammar*. Stanford, CA: CSLI Publications.

Andrew Kehler. 2015. On QUD-based licensing of strict and sloppy ambiguities. In *Semantics and Linguistic Theory (SALT)*, pages 512–532.

Andrew Kehler and Hannah Rohde. 2017. Evaluating an expectation-driven question-under-discussion model of discourse interpretation. *Discourse Processes*, 54(3):219–238.

Sanghee J Kim and Ming Xiang. 2024. Incremental discourse-update constrains number agreement attraction effect. *Cognitive Science*, 48(9):e13497.

Sanghee J Kim, Lang Yu, and Allyson Ettinger. 2022. "No, they did not": Dialogue response dynamics in pre-trained language models. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 863–874, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Todor Koev. 2022. *Parenthetical meaning*. Oxford Studies in Semantics and Pragmatics. Oxford: Oxford University Press.

Margaret Kroll and Matthew W. Wagers. 2019. Working memory resource allocation is not modulated by clausal discourse status. Unpublished manuscript, University of California, Santa Cruz.

Tatsuki Kuribayashi, Yohei Oseki, and Timothy Baldwin. 2024. Psychometric predictive power of large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1983–2005, Mexico City, Mexico. Association for Computational Linguistics.

Marta Kutas, Katherine A DeLong, and Nathaniel J Smith. 2011. A look around at what lies ahead: Prediction and predictability in language processing. In Moshe Bar, editor, *Predictions in the Brain: Using our Past to Generate a Future*. New York, NY: Oxford University Press.

Sol Lago, Diego E Shalom, Mariano Sigman, Ellen F Lau, and Colin Phillips. 2015. Agreement attraction in spanish comprehension. *Journal of Memory and Language*, 82:133–149.

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.

Anna Laurinavichyute and Titus von der Malsburg. 2024. Agreement attraction in grammatical sentences and the role of the task. *Journal of Memory and Language*, 137:104525.

David Lewis. 1979. Scorekeeping in a language game. *Journal of Philosophical Logic*, 8:339–359.

Richard L Lewis and Shravan Vasishth. 2005. An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, 29(3):375–419.

Tal Linzen and Marco Baroni. 2021. Syntactic structure from deep learning. *Annual Review of Linguistics*, 7(Volume 7, 2021):195–212.

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.

Andrew McInnerney and Emily Atkinson. 2020. Syntactically unintegrated parentheticals: Evidence from agreement attraction. The 33rd Annual CUNY Human Sentence Processing, University of Massachusetts Amherst: Amherst, MA. March 19–21 (oral presentation).

Kanishka Misra and Kyle Mahowald. 2024. Language models learn rare phenomena from less rare phenomena: The case of the missing AANNs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 913–929, Miami, Florida, USA. Association for Computational Linguistics.

Shota Momma and Colin Phillips. 2018. The relationship between parsing and generation. *Annual Review of Linguistics*, 4(1):233–254.

Anne Ng and Matthew Husband. 2017. Interference effects across the at-issue/not-at-issue divide: Agreement and NPI licensing. The 30th Annual CUNY Human Sentence Processing, MIT: Cambridge, MA. March 30–April 1 (poster presentation).

Byung-Doh Oh and William Schuler. 2023. Why does surprisal from larger transformer-based language models provide a poorer fit to human reading times? *Transactions of the Association for Computational Linguistics*, 11:336–350.

Byung-Doh Oh, Shisen Yue, and William Schuler. 2024. Frequency explains the inverse correlation of large language models' size, training data amount, and surprisal's fit to reading times. *arXiv preprint arXiv:2402.02255*.

Lalchand Pandia, Yan Cong, and Allyson Ettinger. 2021. Pragmatic competence of pre-trained language models through the lens of discourse connectives. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 367–379, Online. Association for Computational Linguistics.

Dan Parker and Adam An. 2018. Not all phrases are equally attractive: Experimental evidence for selective agreement attraction effects. *Frontiers in Psychology*, 9:1566.

Livia Polanyi. 1988. A formal model of the structure of discourse. *Journal of Pragmatics*, 12(5–6):601–638.

Christopher Potts. 2005. *The logic of conventional implicatures*. Oxford: Oxford University Press.

Christopher Potts. 2012. Conventional implicature and expressive content. In Claudia Maienborn, Klaus von Heusinger, and Paul Portner, editors, *Semantics: An international handbook of natural language meaning*, volume 3, pages 2516–2536. Berlin: Mouton de Gruyter.

Craige Roberts. 2012. Information structure: Towards an integrated formal theory of pragmatics. *Semantics and Pragmatics*, 5(6):1–69.

Hannah Rohde and Andrew Kehler. 2014. Grammatical and information-structural influences on pronoun production. *Language, Cognition and Neuroscience*, 29(8):912–927.

Laura Ruis, Akbir Khan, Stella Biderman, Sara Hooker, Tim Rocktäschel, and Edward Grefenstette. 2024. The goldilocks of pragmatic understanding: fine-tuning strategy matters for implicature resolution by llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Soo Hyun Ryu and Richard L Lewis. 2021. Accounting for agreement phenomena in sentence comprehension with transformer language models: Effects of similarity-based interference on surprisal and attention. *arXiv preprint arXiv:2104.12874*.

Zoe Schlueter, Dan Parker, and Ellen Lau. 2019. Error-driven retrieval in agreement attraction rarely leads to misinterpretation. *Frontiers in Psychology*, 10:1002.

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.

Patrick Sturt, Anthony J Sanford, Andrew Stewart, and Eugene Dawydiak. 2004. Linguistic focus and good-enough representations: An application of the change-detection paradigm. *Psychonomic Bulletin & Review*, 11(5):882–888.

Michelle Suijkerbuijk, Naomi T Shapiro, Peter de Swart, and Stefan L Frank. 2024. The need for human data when analysing the human-likeness of syntactic representations in neural language models: The case of english wh-island constraints.

Kristen Syrett and Todor Koev. 2015. Experimental evidence for the truth conditional contribution and shifting information status of appositives. *Journal of Semantics*, 32(3):525–577.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. LaMBDA: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

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.

Marten van Schijndel and Tal Linzen. 2021. Single-stage prediction models do not explain the magnitude of syntactic disambiguation difficulty. *Cognitive Science*, 45(6):e12988.

Matthew W Wagers, Ellen F Lau, and Colin Phillips. 2009. Agreement attraction in comprehension: Representations and processes. *Journal of Memory and Language*, 61(2):206–237.

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.

Ethan Gotlieb Wilcox, Michael Hu, Aaron Mueller, Tal Linzen, Alex Warstadt, Leshem Choshen, Chengxu Zhuang, Ryan Cotterell, and Adina Williams. 2024. *Bigger is not always better: The importance of human-scale language modeling for psycholinguistics*.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. TransferTransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*.

Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. 2020. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929, Online. Association for Computational Linguistics.

Aditya Yedetore and Najoung Kim. 2024. Semantic training signals promote hierarchical syntactic generalization in transformers. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4059–4073, Miami, Florida, USA. Association for Computational Linguistics.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

A Cross-Genre Analysis of Discourse Relation Signaling in the GUM Corpus

Lauren Levine
Georgetown University
lel76@georgetown.edu

Abstract

In this paper, we investigate the cross-genre variation in how discourse relations are signaled in the Georgetown University Mutilayer (GUM) Corpus, an English language corpus which contains 16 different genres of texts with various linguistic annotations, including Rhetorical Structure Theory (RST) style discourse annotations. We look at the proportions of discourse signals in each genre, and then we conduct an analysis of which discourse relations display the most inter-genre variation in how they are signaled, providing a methodology for ranking the inter-genre variability of the signaling of individual discourse relations. Although the way in which individual discourse relations are signaled in GUM is relatively stable across genres, we are able still to produce stable rankings, finding that organization, restatement, and explanation relations display the most inter-genre variation. However, we find that genre specific graphical norms can account for a large portion of the observed variation.

1 Introduction

Discourse relations are used to describe the meaning that arises from the combination of multiple linguistic units in a discourse. In computational discourse analysis, there have been multiple linguistic formalisms proposed regarding how to annotate this phenomenon, each with their own unique inventories of discourse relations. One such prominent formalism is Rhetorical Structure Theory (RST; Mann and Thompson, 1988), which assigns relation labels on a pragmatic basis, without reference to particular linguistic signals. Despite this, previous work on signaling in RST data has found that over 90% of RST discourse relations are signaled in some way (Das and Taboada, 2018b). This includes overt discourse markers, as shown in Figure 1 with the explicit discourse marker *because*, as well as other more implicit discourse signals. As such, we many wonder if there are patterns in the distributions of which types of discourse signals appear with different discourse relations, and if so, how these patterns appear across different genres. Answers to such questions will provide insight into whether a pragmatic formalism like RST also displays structural patterns in its annotation, which would not necessarily be expected as there are not sturctural criteria in the annotation of RST discourse relations. Such investigation will also provide insights for describing genre differences, particularly regarding how genres use different structural means to achieve a particular discourse purpose.

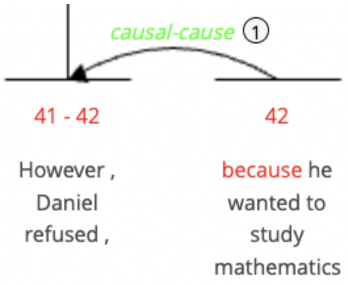

causal-cause
1
41 - 42
42
However , Daniel refused ,
because he wanted to study mathematics

Figure 1: Example from the GUM corpus of a causal discourse relation, overtly signaled by the explicit discourse marker (dm) *because*.

In this paper, we see that different RST relations do in fact co-occur with different proportions of discourse signal types (Figure 2), and we focus in on the question of how different genres signal the same discourse relations. We then consider which individual RST relations display the most inter-genre variation in how they are signaled. In order to investigate this, we introduce an inter-genre variation ranking metric: average pairwise Jensen-Shannon distance (Avg. Pairwise JSD), the details of which are given in Section 4. After we use this metric to obtain an inter-genre variation ranking for different discourse relations, we explore the relations with the most variability to see what this can tell us about the characteristics of different genres.

Overall, we find that while there is clearly inter-relation variation in the distributions of discourse signals, the means of signaling individual RST relations remain relatively consistent across genres. This indicates a general stability in the manner of signaling individual discourse relations, which we would not necessarily expect considering that RST is a pragmatic formalism. However, by utilizing the Avg. Pairwise JSD metric, we are still able to produce stable rankings for which discourse relations show the most cross-genre variation how they are signaled, finding that `organization`, `restatement`, and `explanation` relations display the most inter-genre variation, and that `evaluation` and `adversative` relations show the least inter-genre variation. Code and visualizations for this paper are available on GitHub[1].

2 Previous Work

While relation signaling in the RST formalism is a relatively new area of interest, there are several foundational works which we draw upon in this investigation. Firstly, a major resource for RST data in English is the RST Discourse Treebank (RST-DT), which consists of 385 Wall Street Journal articles (Carlson et al., 2002). In 2013, Taboada and Das subsequently added an additional layer of signaling annotations to a portion of this corpus, and later the entire RST-DT corpus, creating the RST Signalling Corpus (RST-SC; Das and Taboada, 2018a). This work provided the first available RST data with signaling information, and established a manageable taxonomy of signal types, including not only overt discourse markers, but various implicit discourse signals as well. Since its creation, the RST-SC has been used for various corpus analyses of relation signaling (Das and Taboada, 2018b; Das, 2019; Egg and Das, 2022).

There have also been a number of efforts aimed at extending the application of the relation signaling framework created by Taboada and Das. As RST-SC does not indicate which tokens are aligned with the signal type annotations, Liu and Zeldes (2019) made efforts to anchor signaling information directly to tokens in a text. Additionally, Gessler et al. (2019) created an online annotation tool for adding signaling information directly onto existing RST annotations. Both of these efforts were further leveraged in the creation of signaling annotations in data for Enhanced Rhetorical Structure Theory (eRST), an extension of the theoretical RST framework which added a means to account for "tree-breaking, nonprojective and concurrent relations" in discourse relation graphs (Zeldes et al., 2024). The eRST project follows the relation signaling taxonomy from Taboada and Das, dividing relation signals into the following categories: discourse markers, graphical, lexical, morphological, numerical, reference, semantic, and syntactic. The corpus analysis we conduct in this paper is focused on the signaling annotations added to the GUM RST treebank from the eRST project.

3 Data

For this investigation, we use GUM Version 10[2], a 228k token corpus of English, which is composed of 235 documents, divided approximately evenly across 16 different genres: academic, biographies, courtroom, conversation, essay, fiction, interview, letters, news, podcasts, speeches, textbooks, travel, vlogs, how-to and Reddit forum discussions (Zeldes, 2017). As mentioned in the previous section, the GUM corpus has signaling annotations consistent with the form established for the eRST formalism, extended from the taxonomy created by Taboada and Das.

For this analysis, we only consider discourse relations which co-occur with at least one signal annotation (at all levels of the eRST tree). There are a total of 30,774 discourse relation annotations in GUM v10, 69.35% of which (21,343 instances) occur with one or more signaling annotation. The eRST annotations in GUM leverage a two-tiered relation inventory, where the coarse relation and the fine-grained subtype are connected with "-" (e.g., `causal` is the coarse relation type for the fine-grained relation `causal-cause`). The full relation inventory of 15 coarse relations and 32 fine-grained relations is shown in Appendix A. For each relation signal annotation, we extract the signal type and signal subtype, the RST relation type and RST relation subtype (e.g., `elaboration` and `elaboration-attribute`), and the genre in which it occurs from the GUM corpus. This means that a single relation will be extracted multiple times if it occurs with multiple signals. And while we extract both the signal type and the signal subtype, in order

[1] https://github.com/lauren-lizzy-levine/gum-genre-signaling

[2] https://github.com/amir-zeldes/gum

to have enough instances in each signal category to analyze statistically, we limit our investigation to the higher level signal types: discourse markers (dm), graphical (grf), lexical (lex), morphological (mrf), numerical (num), reference (ref), semantic (sem), and syntactic (syn). For reference, the complete signal inventory from Zeldes et al. (2024) is included in Appendix A. For RST discourse relations, we investigate at the level of both coarse relations (e.g., `elaboration`) and fine-grained relations (e.g., `elaboration-attribute`) from the RST relation inventory.

4 Methods

In order to investigate the inter-genre variability of signaling for individual relations, we need a means of quantifying how different the distributions of relation signals are between a pair of genres for a particular relation. We adopt the Jensen-Shannon Distance[3] as metric for this purpose.

The Jensen-Shannon Divergence (JS-Div) is a symmetric measure of the similarity of two probability distributions. This metric is bounded, $0 \leq$ JS-Div ≤ 1, where 0 indicates the distributions are identical, and 1 indicates they are completely different. The Jensen-Shannon Distance (JSD) is the square root of the JS-Div, and it is commonly used to assess the similarity of probability distributions. In order to apply JSD as a metric to our relation signaling data, we make the assumption that the frequency counts of the signal types used to indicate a relation in a specific genre can be used to approximate the probability distribution of how that relation is signaled in that genre[4].

For each relation (e.g., `explanation`), this gives us per genre probability distributions of signal types which we can compare using JSD. We can then calculate the JSD scores between all possible pairs of genres (e.g., ('reddit', 'academic'): 0.63, ('interview', 'academic'): 0.59, etc.). We use these scores for two purposes: First, we construct a distance matrix for genre pairs which can be used as input for clustering/dendrograms of genre similarity (with respect to signaling). Secondly, we can take an average of these scores to create a single number that represents the inter-genre variability score for the individual relation (e.g., 0.35). We refer to this metric for quantifying inter-domain variation as the average pairwise Jensen-Shannon Distance (Avg. Pairwise JSD). We note that while in this study we specifically use the metric to investigation the inter-genre variation in how individual discourse relations are signaled, it can be thought of as a more general metric. Genre, relation, and signal type may be swapped out for other categories as the context requires.

Rank Correlation Metric	Relation Type	
	Coarse	Fine-grained
Avg. Kendall's Tau	0.82	0.76
Avg. Spearman Rank	0.93	0.90
Avg. Pearson Correlation	0.95	0.92

Table 1: Averages of correlation metrics from comparing rankings of inter-genre variation for the signaling of individual RST relations, computed from randomly sampled subsets of the GUM corpus.

The inter-genre variability score for a discourse relation R using Avg. Pairwise JSD is defined as:

$$Avg.PairwiseJSD(R) =$$

$$\frac{\sum_{i,j \in G} JSD(SD_i(R), SD_j(R))}{\binom{|G|}{2}}$$

where G is the set of genres, JSD is the Jensen-Shannon Distance, and $SD_x(R)$ is the frequency distribution of relation signal types for relation R in genre x.

For the fine-grained relations, the frequency distribution of the relation signal types is approximated by the raw frequency counts in the data. For the coarse relations, we normalize the frequency distribution by the proportions of sub-relations composing the coarse relation. We treat each sub-relation as an independent class within the coarse relation, and we take the macro-average of the distributions for the individual classes to be frequency distribution for the coarse relation.

Once the Avg. Pairwise JSD scores are calculated, they can be sorted to give a relative ranking of inter-genre variability of signaling amongst individual relations. In order to establish how reliably the Avg. Pairwise JSD is able to construct this relative ranking, we compare the rankings that this methodology produces when applied to different subsets of the data. For each genre, we randomly sample 5 documents, and we compute the relative rankings of inter-genre variability as described

[3] https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.jensenshannon.html

[4] This assumption means that you take the frequency counts to be a representative distribution of the categories of signal types. This may not be a valid assumption if the data is sparse.

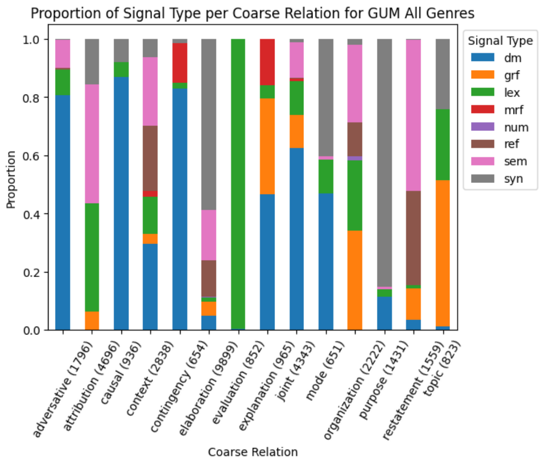


Figure 2: Proportions of relation signal types for coarse RST relations in the GUM corpus. The total number of occurrences of a relation type co-occuring with a signal is included in parentheses.

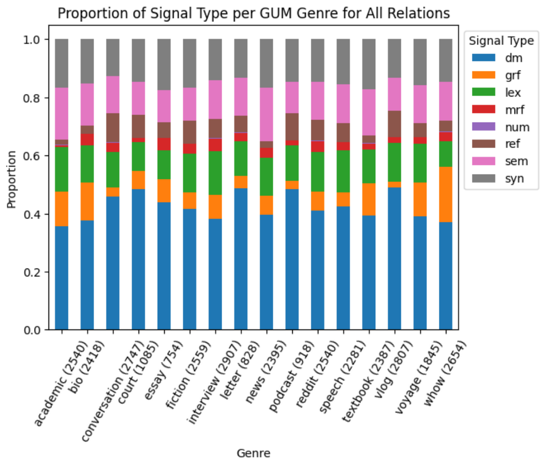


Figure 3: Proportions of relation signal types for the genres in the GUM corpus. The total number of occurrences of relations co-occuring with a signal in the given genre is included in parentheses.

above. We repeat this process 50 times, so there are 50 independent rankings (each with Avg. Pairwise JSD scores) for both coarse and fine-grained RST relations. For each pair of rankings in this 50 run sequence (1225 pairs), we calculate the following correlation metrics between the rankings: Kendall's Tau, Spearman Rank and Pearson Correlation, and then we average the resulting scores for each.

We report the averages for the rank correlation metrics Table 1. For all of these metrics, the closer the score is to 1, the closer the correspondence between the rankings/scores being compared. We see that all the metrics are quite high, and that the metrics for coarse relation ranking averages are higher than those for the fine-grained relation rankings. The strength of the correlation coefficients shows that the rankings are relatively stable, even when data is randomly sampled. We take this to be a reasonable indication that Avg. Pairwise JSD can be used to reliably construct a relative ranking of inter-genre variation for individual relations.

5 Results

To begin our analysis, we investigate the variation in signal types used for different relations in the GUM corpus. Figure 2 visualizes the proportions of signal types used with each coarse RST relation in the GUM corpus. We see that there is a considerable amount of inter-relation variation, and there are some interesting observations to be made from this visualization alone: `evaluation` relations are signaled exclusively by lexical features, `adversative`, `causal`, and `contingency` relations are dominated by overt discourse markers, etc.

However, as this investigation is focused on the inter-genre variation of individual relations, we shift our focus to explore the distribution of relations signals across the different genres in the GUM corpus. We provide a visualization for this analysis in Figure 3, which shows the proportions of signals present in each genre, adjusted for the relative frequencies of the relations present in that genre[5].

In Figure 3, we see that the signal proportions are surprisingly consistent across the various genres of the GUM corpus. We face the possibility that individual relations do not display a substantial amount of inter-genre variation overall, and, as such, we need to focus in on the areas of our data which display the most inter-genre variation for investigation. To this end, we create a relative ranking of the inter-genre variability of signaling amongst individual relations via the methods described in Section 4. The inter-genre variation ranking for the coarse RST relations is shown in Figure 4, and the inter-genre variation ranking for the fine-grained RST relations is shown in Figure 5.

Looking at the ranking for coarse rela-

[5] The signal type proportions for each fine-grained relation attested in the genre are calculated separately and each one considered a separate class. The macro-average of these classes is then taken and reported in Figure 3 as the signal distribution of the genre.

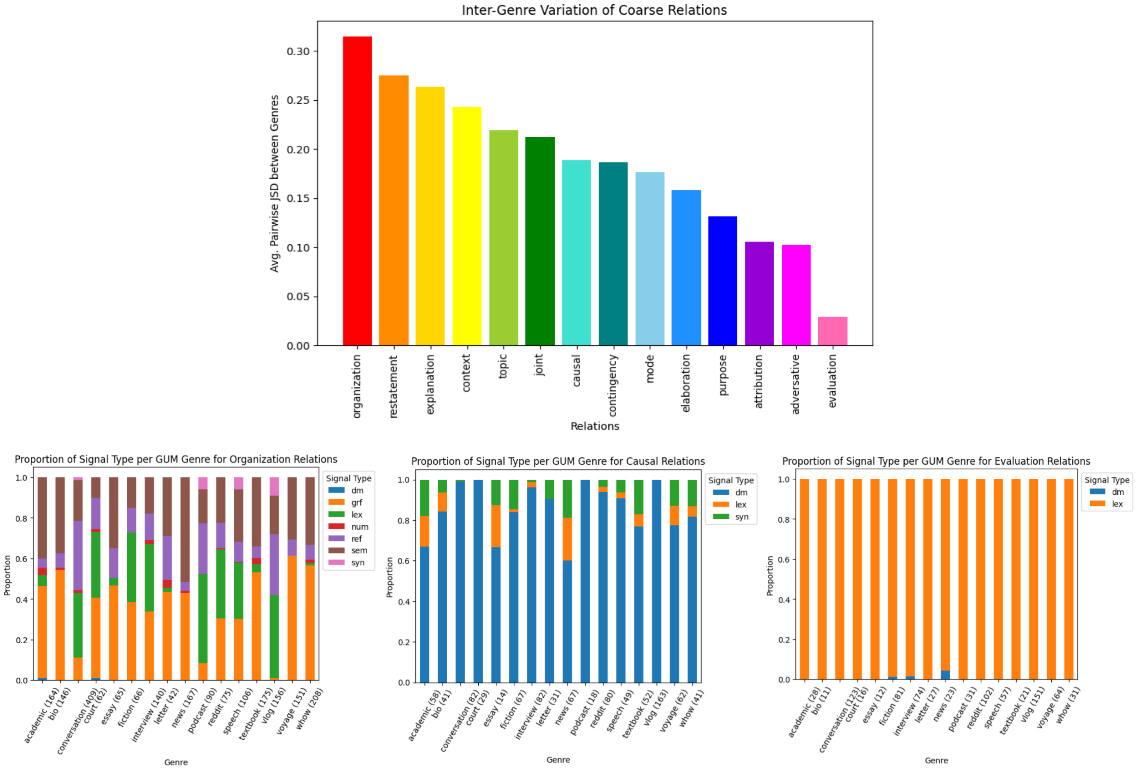


Figure 4: (Top) Ranking of inter-genre variation of relation signal distributions for coarse RST relations (based on Avg. Pairwise JSD). Proportions of relation signals across genres for: (bottom left) the coarse relation showing the most variation: `organization`, (bottom middle) the coarse relation showing the median variation: `causal`, and (bottom right) the coarse relation showing the least variation: `evaluation`.

tions in Figure 4, we see that `organization`, `restatement`, and `explanation` relations display the most inter-genre variation, while `attribution`, `adversative`, and `evaluation` relations display the least inter-genre variation. In the bottom section of Figure 4, we also show the signal type distributions across genres for the relations whose Avg. Pairwise JSD indicated that they show the most (`organization`), median (`causal`), and lowest (`evaluation`) inter-genre variation. As we can see from the visualizations, Avg. Pairwise JSD seems to accurately reflect the relative inter-genre variation of the relations.

Looking at the ranking for fine-grained relations in Figure 5, we see that `explanation-evidence`, `restatement-partial`, and `restatement-repetition` relations display the most inter-genre variation, while `elaboration-attribute`, `explanation-justify`, and `evaluation-comment` relations display the least inter-genre variation. In the bottom section of Figure 5, we again show the signal type distributions across genres for the relations whose Avg. Pairwise JSD indicated that they show the most (`explanation-evidence`), median (`adversative-antithesis`), and lowest (`evaluation-comment`) inter-genre variation. As we can again see from these visualizations, Avg. Pairwise JSD accurately reflects the relative inter-genre variation of the relations.

Now that we have rankings of the inter-genre variability for relations, we will take a look at some of the individual relations which displayed the most variation: `organization` and `explanation`. First, consider Figure 6. The left side of the figure shows the distribution of relation signal types across genres for the `organization` relation. The right side of the figure is a dendrogram showing the signaling similarity between genres for the `organization` relation (based on a distance matrix of JSD scores between genre pairs).

Looking at the dendrogram in Figure 6, we see that there is a relatively clear split between spoken genres and written genres. This means, perhaps unsurprisingly, that written genres and spoken genres are relatively distinct in how they signal

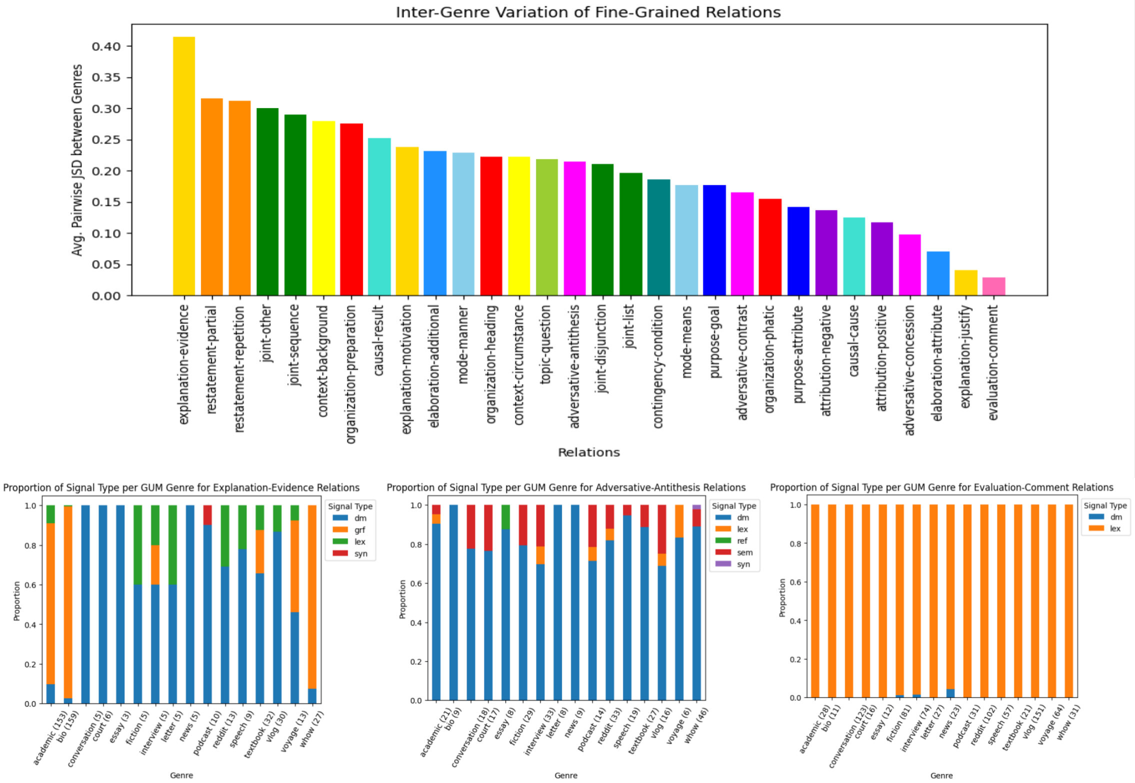


Figure 5: (Top) Ranking of inter-genre variation of relation signal distributions for fine-grained RST relations (based on Avg. Pairwise JSD). Proportions of relation signals across genres for: (bottom left) the fine-grained relation showing the most variation: explanation-evidence, (bottom middle) the fine-grained relation showing the median variation: adversative-antithesis, and (bottom right) the fine-grained relation showing the least variation: evaluation-comment.

organization relations. If we look at the graph on the left side of Figure 6, we see that spoken genres, such as conversation and podcast, have a much smaller proportion of graphical signals than the written genres. This is intuitive, as there are many graphical signals, such as headings, that are commonly used in written genres for organizational purposes, which cannot be used in spoken genres. Instead, we see that the lexical signal type compensates for the lack of graphical signals in spoken genres.

Now consider Figure 7. On the left, we have the distribution of relation signal types across genres for the explanation relation. On the right, we have a genre signaling similarity dendrogram, this time for the explanation relation. In this dendrogram, we can see that there is a clear split between academic, biographies and wiki-how, and the rest of the genres. If we look at the graph on the left of Figure 7, we once again see that different proportions of graphical signals are largely responsible for this divergence. Upon qualitative examination of the data, we see that this is largely due to parentheses being used for citations, a practice which is common in academic writing, biographies, and wiki articles.

6 Discussion

In the results of our investigation, we saw that the inter-genre signaling of individual discourse relations is relatively stable. In two of the coarse relations which showed the most inter-genre variation in their signaling, organization and explanation, genre specific graphical norms seemed to contribute more to the existing variation than the language content. As such, if there is a large variation in the signal types used in two genres that goes beyond graphical norms, it may be because those genres call for different relations to be used, rather than because the genre is signaling the same relations differently.

It is somewhat surprising that we see such limited variation in the signaling of individual relations

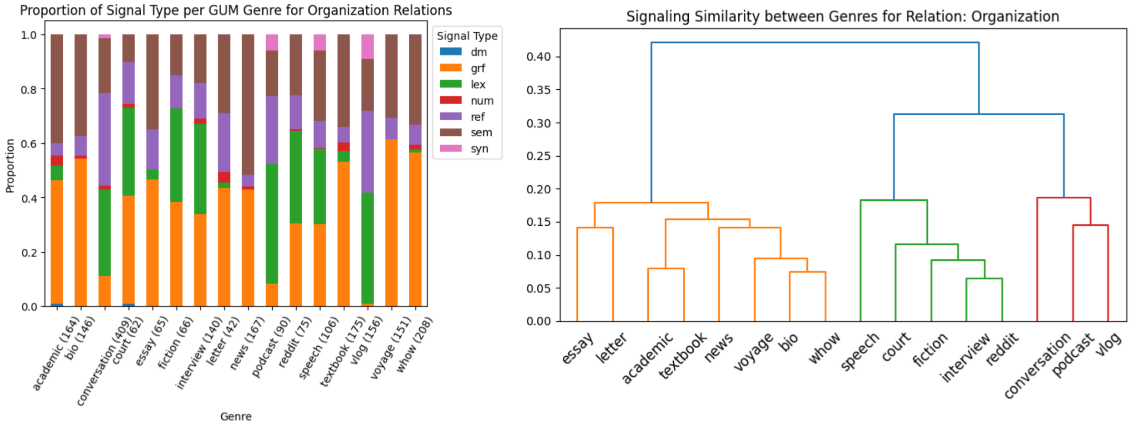


Figure 6: (Left) Proportions of relation signal types across genres for the `organization` relation. (Right) Dendrogram showing the signaling similarity between genres for the `organization` relation.

across genres, particularly considering that RST is a pragmatic formalism, and thus does not have restrictions on the structural components that must be present in order apply a specific discourse relation. Our results suggests that, despite being pragmatically defined, the discourse relations in the RST relation inventory display some degree of structural consistency in their manner of signaling. However, it is also worth noting that many of the signaling annotations from the GUM corpus which we are analyzing were automatically annotated by NLP tools/scripts. These automatic processes rely on restrictive heuristics, which may artificially limit the signaling variation being captured by the annotations. In future work, it would be beneficial to consider the specific limitations being imposed by such automatic annotations.

7 Conclusion

In this paper, we explored the cross-genre variation in how discourse relations are signaled in the GUM Corpus. We looked at the proportions of discourse signals in each genre, and we saw that there is a relative stability in how discourse relations are signaled across genres. We then conducted an analysis of which discourse relations display the most inter-genre variation in how they are signaled, using as a pairwise average of the JSD scores between different genres (Avg. Pairwise JSD) a metric of the inter-genre variability of individual discourse relations. We found that `organization`, `restatement`, and `explanation` relations display the most inter-genre variation, and that `evaluation` and `adversative` relations show the least inter-genre variation. Amongst the relations displaying the most inter-genre variation, we saw that the divide between spoken genres and written genres, and the accompanying divergence in graphical norms between the two modalities, is salient in accounting for the observed variation. Overall, we found that the RST discourse relations in GUM are signaled in a relatively stable manner across genres, and that the variation that does exist seem to largely come from differences in graphical norms, rather than differences in linguistic content.

Limitations

As noted in Section 4, using JDS as a metric for inter-genre variation relies on there being enough data to satisfy the assumption that the frequency of occurrence of signals is representative for the way that a relation is signaled in that genre. However, not all of the genres in the GUM corpus have the same number of documents, and for those with less documents, such as essay, which only has 5 documents, it is less sure that the assumption is sound. Still, the results from our correlation metrics in Section 4 in suggest that 5 documents is sufficient to give a reasonably stable ranking.

Additionally, as noted the Section 6, many of the signal annotations in the GUM corpus were automatically generated with NLP tools/scripts, which may limit the observable degree of inter-genre variation for relation signaling. A greater understanding of the inter-genre variation for relation signaling could be had from looking at a larger number of manual annotations, or by better accounting for the biases introduced by the automatic annotation tools.

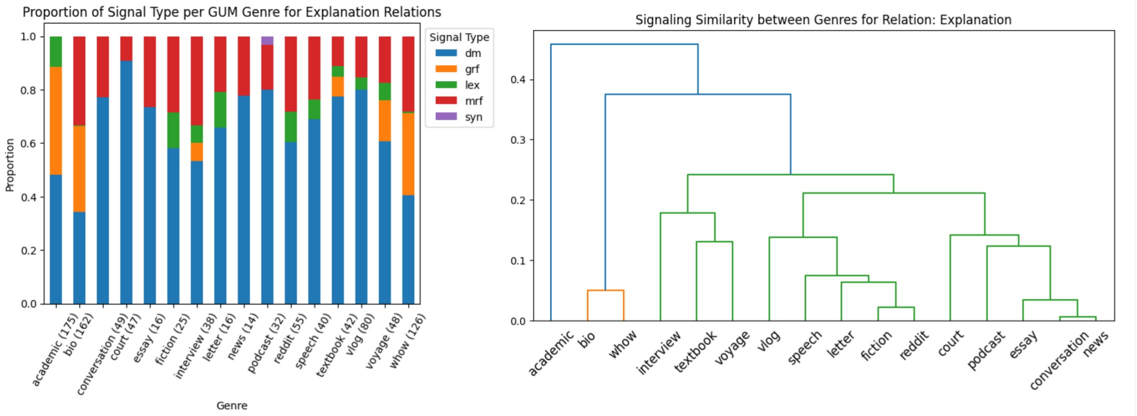


Figure 7: (Left) Proportions of relation signal types across genres for the `explanation` relation. (Right) Dendrogram showing the signaling similarity between genres for the `explanation` relation.

References

Lynn Carlson, Mary Ellen Okurowski, and Daniel Marcu. 2002. RST discourse treebank. *Linguistic Data Consortium, University of Pennsylvania.*

Debopam Das. 2019. Nuclearity in RST and signals of coherence relations. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 30–37, Minneapolis, MN. Association for Computational Linguistics.

Debopam Das and Maite Taboada. 2018a. RST signalling corpus: A corpus of signals of coherence relations. *Language Resources and Evaluation*, 52:149–184.

Debopam Das and Maite Taboada. 2018b. Signalling of coherence relations in discourse, beyond discourse markers. *Discourse Processes*, 55(8):743–770.

Markus Egg and Debopam Das. 2022. Signalling conditional relations. *Linguistics Vanguard*, 8(s4):383–392.

Luke Gessler, Yang Janet Liu, and Amir Zeldes. 2019. A discourse signal annotation system for RST trees. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 56–61.

Yang Liu and Amir Zeldes. 2019. Discourse relations and signaling information: Anchoring discourse signals in RST-DT. *Society for Computation in Linguistics*, 2(1).

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Maite Taboada and Debopam Das. 2013. Annotation upon annotation: Adding signalling information to a corpus of discourse relations. *Dialogue & Discourse*, 4(2):249–281.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Amir Zeldes, Tatsuya Aoyama, Yang Janet Liu, Siyao Peng, Debopam Das, and Luke Gessler. 2024. eRST: A signaled graph theory of discourse relations and organization. *arXiv preprint arXiv:2403.13560*.

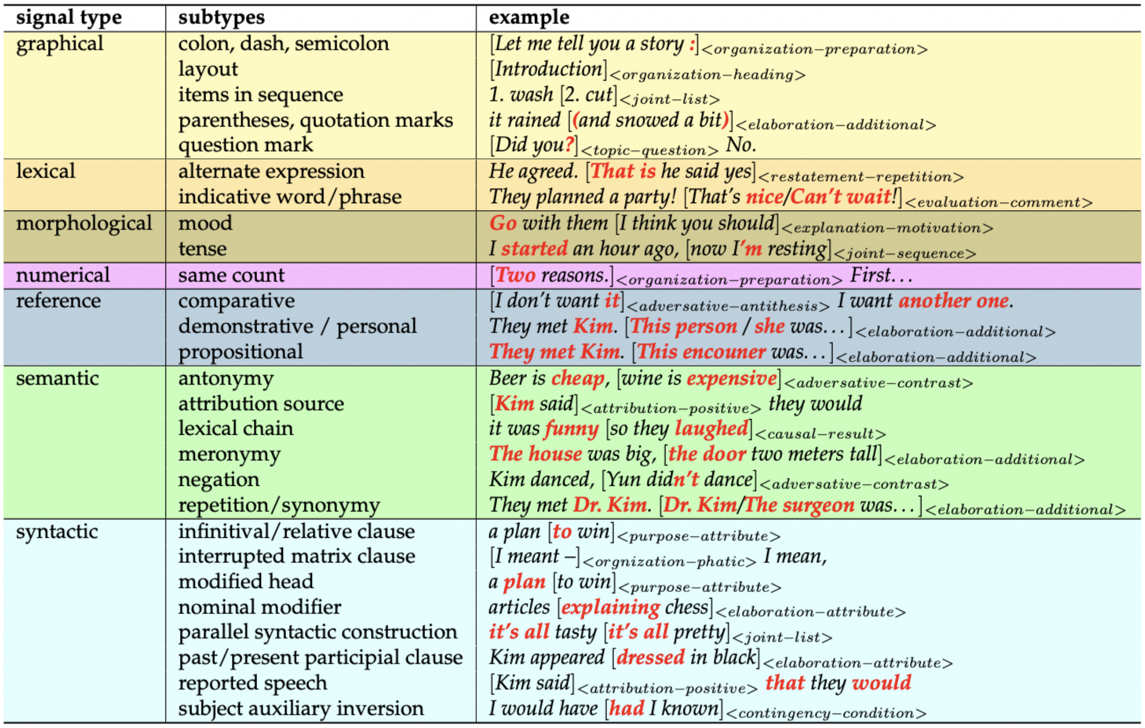
A RST Relation Inventory and eRST Signal Inventory in GUM v10

In Table 2 we include the RST relation inventory used in GUM v10, listing both coarse and fine-grained relations. For reference, in Figure 8 we include the eRST signaling inventory presented in Zeldes et al. (2024).

RST Relation Inventory

Coarse	Fine-grained	Coarse	Fine-grained
ADVERSATIVE	ADVERSATIVE-ANTITHESIS ADVERSATIVE-CONCESSION ADVERSATIVE-CONTRAST	JOINT	JOINT-DISJUNCTION JOINT-LIST JOINT-SEQUENCE
ATTRIBUTION	ATTRIBUTION-POSITIVE		JOINT-OTHER
	ATTRIBUTION-NEGATIVE	MODE	MODE-MANNER
CAUSAL	CAUSAL-CAUSE		MODE-MEANS
	CAUSAL-RESULT	ORGANIZATION	ORGANIZATION-HEADING
CONTEXT	CONTEXT-BACKGROUND CONTEXT-CIRCUMSTANCE		ORGANIZATION-PHATIC ORGANIZATION-PREPARATION
CONTINGENCY	CONTINGENCY-CONDITION	PURPOSE	PURPOSE-ATTRIBUTE
ELABORATION	ELABORATION-ATTRIBUTE		PURPOSE-GOAL
	ELABORATION-ADDITIONAL	RESTATEMENT	RESTATEMENT-PARTIAL
EXPLANATION	EXPLANATION-EVIDENCE		RESTATEMENT-REPETITION
	EXPLANATION-JUSTIFY EXPLANATION-MOTIVATION	TOPIC	TOPIC-QUESTION TOPIC-SOLUTIONHOOD
EVALUATION	EVALUATION-COMMENT	SAME-UNIT	SAME-UNIT

Table 2: RST Relation Inventory in GUM v10.



signal type	subtypes	example
graphical	colon, dash, semicolon	*[Let me tell you a story :]*$_{<organization-preparation>}$
	layout	*[Introduction]*$_{<organization-heading>}$
	items in sequence	*1. wash [2. cut]*$_{<joint-list>}$
	parentheses, quotation marks	*it rained [(and snowed a bit)]*$_{<elaboration-additional>}$
	question mark	*[Did you?]*$_{<topic-question>}$ *No.*
lexical	alternate expression	*He agreed. [That is he said yes]*$_{<restatement-repetition>}$
	indicative word/phrase	*They planned a party! [That's nice/Can't wait!]*$_{<evaluation-comment>}$
morphological	mood	*Go with them [I think you should]*$_{<explanation-motivation>}$
	tense	*I started an hour ago, [now I'm resting]*$_{<joint-sequence>}$
numerical	same count	*[Two reasons.]*$_{<organization-preparation>}$ *First...*
reference	comparative	*[I don't want it]*$_{<adversative-antithesis>}$ *I want another one.*
	demonstrative / personal	*They met Kim. [This person / she was...]*$_{<elaboration-additional>}$
	propositional	*They met Kim. [This encouner was...]*$_{<elaboration-additional>}$
semantic	antonymy	*Beer is cheap, [wine is expensive]*$_{<adversative-contrast>}$
	attribution source	*[Kim said]*$_{<attribution-positive>}$ *they would*
	lexical chain	*it was funny [so they laughed]*$_{<causal-result>}$
	meronymy	*The house was big, [the door two meters tall]*$_{<elaboration-additional>}$
	negation	*Kim danced, [Yun didn't dance]*$_{<adversative-contrast>}$
	repetition/synonymy	*They met Dr. Kim. [Dr. Kim/The surgeon was...]*$_{<elaboration-additional>}$
syntactic	infinitival/relative clause	*a plan [to win]*$_{<purpose-attribute>}$
	interrupted matrix clause	*[I meant –]*$_{<orgnization-phatic>}$ *I mean,*
	modified head	*a plan [to win]*$_{<purpose-attribute>}$
	nominal modifier	*articles [explaining chess]*$_{<elaboration-attribute>}$
	parallel syntactic construction	*it's all tasty [it's all pretty]*$_{<joint-list>}$
	past/present participial clause	*Kim appeared [dressed in black]*$_{<elaboration-attribute>}$
	reported speech	*[Kim said]*$_{<attribution-positive>}$ *that they would*
	subject auxiliary inversion	*I would have [had I known]*$_{<contingency-condition>}$

Figure 8: Signal inventory for eRST given in Zeldes et al. (2024): "Non-DM signal types and subtypes, with examples highlighting in red the signal tokens which indicate the relation of the unit in square brackets."

Similarity, Transformation and the Newly Found Invariance of Influence Functions

Andrew Liu
University of Toronto
aliu@cs.toronto.edu

Gerald Penn
University of Toronto
gpenn@cs.toronto.edu

Abstract

Ensuring that semantic representations capture the actual meanings of sentences to the exclusion of extraneous features remains a difficult challenge despite the amazing performance of representations like sBERT. We compare and contrast the semantic-encoding behaviours of sentence embeddings as well as *influence functions*, a resurgent method in the field of language model intepretability, using meaning-preserving grammatical transformations. Under the two tasks of sentence similarity and a new task called *entity invariance*, we seek to understand how these two measures of semantics warp under surface-level syntactic changes. Invariance to meaning-preserving transformations is an important aspect in which sentence embeddings and influence functions seem to differ. Nevertheless, our experiments find that across all our tasks and transformations, sentence embeddings and influence functions are highly correlated. We conclude that there is evidence that influence functions point towards a deeper encoding of semantics.

1 Introduction

A major concern with neural language models is their lack of transparency. In addition to the expense of even functionally observing the predictions of a model, there is the additional concern of *why* it happened. A number of recent attempts at probing or interpreting language-model predictions have relied upon either misbegotten characterizations of linguistic theory in relation to those predictions, or naıve metaphorical proxies for linguistic theory, such as the retrieval of knowledge from a computer's memory, or assigning distributions to sentences as points in a discrete set of outcomes, rather than as points in a continuous, albeit inscrutable, latent semantic space.

A case in point is the resurgence of the notion of an "influence function" (Hampel, 1974), which attempts to assign weight to training sentences that an erroneous, indiscreet or salacious output can then be traced back to. Until very recently, the use of influence functions in LLMs was not computationally feasible. Now that it is somewhat feasible, the question is what it makes sense to do with them. In particular, the authors of these several papers on optimization and approximation of influence functions apparently never considered whether influence was merely a direct consequence of semantic similarity, a topic with a long history of proposed quantitative methods.

The central claim of this paper is that a better understanding of the potential of influence functions is attainable with a slightly less superficial understanding of linguistic theory. In particular, as a complement to the task of directly computing the semantic similarity of two expressions, we introduce the task of *entity invariance*, in which two related sentences are examined relative to a semantic argument that they share. The relation between these two sentences is composed of *grammatical transformations*, a now rather antiquated term for regular, meaning-preserving correspondences (at least in a reading that equates meaning with thematic role assignment) between syntactic forms. Passivization, topicalization and clefting are examples of transformations. (Chomsky, 1965) (Lambrecht, 2001) (Aelbrecht and Haegeman, 2012).

We describe a series of experiments and descriptive hypothesis tests which demonstrate that, under certain conditions, influence functions have a greater potential for invariance to syntactic transformations than conventional sentence embeddings in large-dimensional vector spaces. Just as in computer vision, where the ability to identify a shape is naturally tested for translation and rotation invariance, we assert that a semantic representation should be tested for invariance to diathesis and other syntactic transformations that ostensibly preserve meaning.

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 309-321.
Eugene, Oregon, July 18-20, 2025

2 Methods

2.1 Sentence-BERT

As a canonical example of sentence embeddings, we select all-mpnet-base-v2 (Reimers and Gurevych, 2019), a sentence-transformer model that encodes sentences into a 768-dimensional dense vector space. The underlying model is the Microsoft mpnet-base model, pre-trained with the MPNet objective function (Song et al., 2020):

$$\mathbb{E}_{z\in Z}\sum_{t=c+1}^{n}\log P(x_{z_t}|x_{z_{\leq c}}, M_{z_{>c}};\theta) \qquad (1)$$

This is a unified pre-training objective for both Masked Language Modeling (Devlin et al., 2019) and Permuted Language Modeling (Yang et al., 2019), inheriting the strengths of both. A sequence is permuted, and its right-most tokens are masked. The goal is then to predict the value of the masked token conditioned on all tokens preceding it, $x_{z_{\leq c}}$, and the positions of the other masked tokens $M_{z_{>c}}$.

The model is then contrastively fine-tuned between sentence pairs in batches by computing the cosine similarities of their embeddings and comparing the cross-entropy loss with true pairs. The cosine similarities produce a value from -1 to 1. The cross-entropy loss then encourages the true pairs to have a larger value (closer to 1) while the non-pairs have a smaller one (closer to -1).

The resulting model accepts a sentence or paragraph and produces a vector encoding that captures some semantically relevant information.

2.2 Influence Functions

Influence functions are an older idea from statistics, re-introduced only recently to deep learning (Koh and Liang, 2017). Suppose there is a training dataset $D = \{z_i\}_{i=1}^{N}$ and a model with parameters $\theta \in \mathbb{R}^D$, fit using a loss function $\mathcal{L}$:

$$\theta^* = \arg\min_{\theta\in\mathbb{R}^D}\mathcal{J}(\theta, D) = \arg\min_{\theta\in\mathbb{R}^D}\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(z_i, \theta). \qquad (2)$$

With this, we would like to investigate the effect of adding or removing a single training example z_m on the optimal parameters θ^*. By weighting that new training example by ϵ, we can describe the new optimal parameters with an additional training example as:

$$\theta^*(\epsilon) = \arg\min_{\theta\in\mathbb{R}^D}\mathcal{J}(\theta, D_\epsilon) \qquad (3)$$

$$= \arg\min_{\theta\in\mathbb{R}^D}\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(z_i, \theta) + \epsilon\mathcal{L}(z_m, \theta). \qquad (4)$$

Influence is defined as the first-order Taylor approximation to this function evaluated at $\epsilon = 0$. Using the Implicit Function Theorem, this is:

$$\mathcal{I}_{\theta^*}(z_m) = -H^{-1}\nabla_\theta\mathcal{L}(z_m, \theta^*) \qquad (5)$$

where $H = \nabla_\theta^2\mathcal{J}(\theta^*, D)$ is the Hessian of the empirical-loss function with the original dataset.

Since $\mathcal{I}_{\theta^*}$ is the linear approximation at 0, we can approximate the change in parameters as follows:

$$\theta^*(\epsilon) - \theta^* \approx \mathcal{I}_{\theta^*}(z_m)\epsilon \qquad (6)$$

$$= -H^{-1}\nabla_\theta\mathcal{L}(z_m, \theta^*)\epsilon \qquad (7)$$

Now, when we set $\epsilon = -\frac{1}{N}$ for some datapoint z_m already in the dataset, this corresponds to the effect of removing that datapoint.

Lastly, a change in parameters is difficult to interpret, so typically influence is measured on a more meaningful quantity such as validation loss or perplexity. Luckily, this can easily be done for any quantity $f(\theta)$ using the chain rule. For any meaningful measure f:

$$\mathcal{I}_f(z_m) = \nabla_\theta f(\theta^*)^T\mathcal{I}_{\theta^*}(z_m) \qquad (8)$$

$$= -\nabla_\theta f(\theta^*)^T H^{-1}\nabla_\theta\mathcal{L}(z_m, \theta^*) \qquad (9)$$

Applying $I_f(z_m)$ in the same way as before, we can approximate the change in this measure f due to the addition/removal of a datapoint with the following:

$$f(\theta^*(\epsilon)) - f(\theta^*) \approx \mathcal{I}_f(z_m)\epsilon \qquad (10)$$

$$= -\nabla_\theta f(\theta^*)^T H^{-1}\nabla_\theta\mathcal{L}(z_m, \theta^*)\epsilon. \qquad (11)$$

2.2.1 Influence in the Domain of LLMs

While influence functions are an old idea, numerous limitations kept them from being practical when examining neural-network-based architectures (Bae et al., 2024) (Zhang and Zhang, 2022) (Basu et al., 2021).

- Loss landscapes are not fully convex, meaning that the Hessian can be singular (thus it has no inverse).

- Even if the loss landscape were convex, the formulation of these objective functions implicitly assumes the model is trained to full convergence, which is almost never the case.

- Even if neither of these were an issue, the task of inverting the Hessian is by itself time-consuming.

These limitations have, over time, been addressed (Martens and Grosse, 2015) (George et al., 2018) (Martens, 2020) (Bae et al., 2024), mainly with clever approximations. The final result is then a reasonably efficient method for calculating influence for analyzing even large language models (Li et al., 2024), which we employ for our experiments. For a more detailed explanation, we refer the reader to (Grosse et al., 2023).

2.2.2 Influence for Language Modeling and Transformers

To use influence on the language-modelling task, we simply set the quantity f to be the following:

$$f(\theta) = \log p(z_c; \theta) \tag{12}$$

where z_c is the model's output and θ are the parameters of the transformer model. We follow previous work and use GPT2 (Radford et al., 2019) as the model to analyze, for which this log-likelihood decomposes using Bayes's Rule. Then the influence function approximates the instantaneous change in log-likelihood of generating an output z_c when removing or adding a piece of training data. For example, when a model generates, "Pythagoras was a ...", the presence of a training datapoint like "the Pythagorean theorem ..." is intuitively more important to this prediction than something less related like "The doctor suggested ...". Influence allows us to quantify the effect of a single datapoint from the training set by ablating it.

3 Problem Description

We investigate two capacities that we conjecture to be desirable of any model that aspires to true semantic reasoning: the now very well-studied ability to calculate the similarity in meaning between two sentences, and an invariance to meaning-preserving syntactic transformations.

In particular, we define *entity invariance* as a three-way comparison in which the congruence of the (now, usually a vector) representation of a fixed referring expression is calculated with a sentence that uses it, but relative to a baseline in which the same congruence is calculated between the same referring expression and a different but closely related sentence. For example, while the precise geometric relationship between the designator *John* and *John threw the ball* may be mostly inscrutable within modern neural vector representations of word and sentence meaning, we are perhaps justified in expecting that this relationship, whatever it is, will be the same as the one between *John* and *The ball was thrown by John*, *The ball, John threw* or *It is the ball that John threw,* because these various transformations are ostensibly meaning-preserving. This is a higher-order alternative to directly calculating the sentence similarity between the representations of *John threw the ball* and *The ball was thrown by John*, viewed through the lens of the meaning of *John*.

This has further implications with respect to phenomena like semantic masking (Shi and Penn, 2025), in which asymmetries have been observed in the ability of a document's context to obscure various inserted passages of text in question-answering tasks with LLMs. Rephrasing under a meaning-preserving transformation can actually alter these effects if the entity answer to a factoid question is not invariant to its sentence location.

The motivation behind both tasks is the same: given some semantics-related task, when replacing a sentence with a semantically equivalent yet syntactically transformed alternative, it should be the case that any method that claims to encode semantics should be robust to this replacement. Essentially, we claim moving across semantics-preserving transformations should not change the behavior of a true measure of semantics. For example, if sentence A is similar to sentence B according to some measure, and A' is the passivized form of A, then A' should be equally similar to sentence B. This is the sentence similarity task. If the subject of sentence A is deemed important by some measure, then the importance of that same subject on the sentence A' should also be equally important by that measure. This is the entity invariance task. We can approach both tasks with the aforementioned semantic tools: cosines of sBERT vectors and influence functions. An example is illustrated in Figure 1.

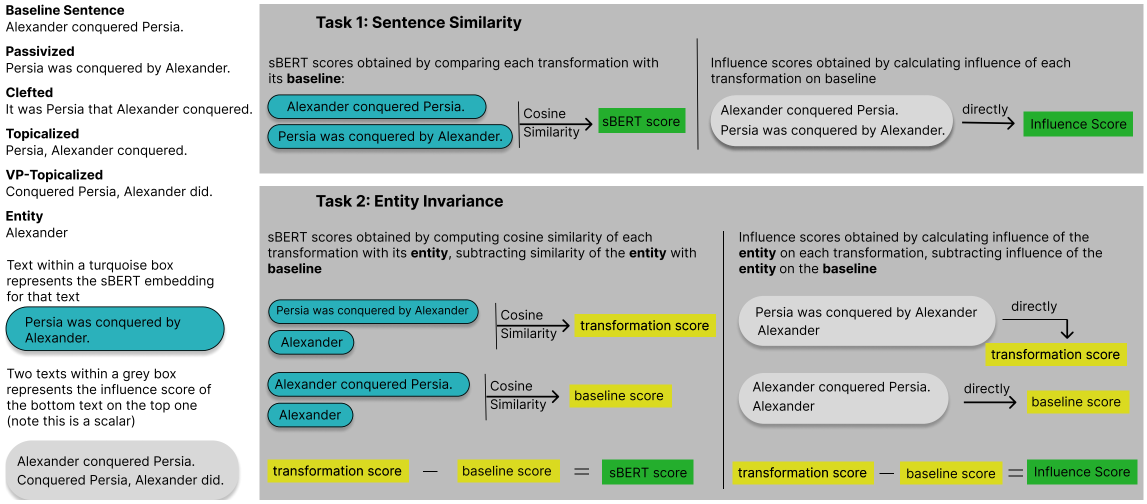

Baseline Sentence
Alexander conquered Persia.
Passivized
Persia was conquered by Alexander.
Clefted
It was Persia that Alexander conquered.
Topicalized
Persia, Alexander conquered.
VP-Topicalized
Conquered Persia, Alexander did.
Entity
Alexander
Text within a turquoise box represents the sBERT embedding for that text
Persia was conquered by Alexander.
Two texts within a grey box represents the influence score of the bottom text on the top one (note this is a scalar)
Alexander conquered Persia.
Conquered Persia, Alexander did.
Task 1: Sentence Similarity
sBERT scores obtained by comparing each transformation with its baseline:
Alexander conquered Persia.
Persia was conquered by Alexander.
Cosine Similarity
sBERT score
Influence scores obtained by calculating influence of each transformation on baseline
Alexander conquered Persia.
Persia was conquered by Alexander.
directly
Influence Score
Task 2: Entity Invariance
sBERT scores obtained by computing cosine similarity of each transformation with its entity, subtracting similarity of the entity with baseline
Persia was conquered by Alexander
Alexander
Cosine Similarity
transformation score
Alexander conquered Persia.
Alexander
Cosine Similarity
baseline score
transformation score — baseline score = sBERT score
Influence scores obtained by calculating influence of the entity on each transformation, subtracting influence of the entity on the baseline
Persia was conquered by Alexander
Alexander
directly
transformation score
Alexander conquered Persia.
Alexander
directly
baseline score
transformation score — baseline score = Influence Score

Figure 1: Example of both tasks under both metrics. Above shows the walkthrough of getting scores for the sentence "Alexander conquered Persia." in the passivized transformation. The above calculations are repeated for each transformation and for each sentence.

Baseline	Alexander conquered Persia.
Passivization	Persia was conquered by Alexander.
Clefting	It was Persia that Alexander conquered.
Topicalization	Persia, Alexander conquered.
VP-Topicalization	Conquered Persia, Alexander did.

Table 1: One entry of the Grammatical Transformation Dataset

Baseline	Zorvik climbed Everest.
Passivization	Everest was climbed by Zorvik.
Clefting	It was Everest that Zorvik climbed.
Topicalization	Everest, Zorvik climbed.
VP-Topicalization	Climbed Everest, Zorvik did.

Table 2: One entry of the Made-Up Entity Dataset

4 Experimental Setup

4.1 Datasets

4.1.1 Sentence Sampling and the Grammatical Transformation Dataset

In order to investigate grammar transformations on our semantic tasks in a controlled manner, we created a dataset that contains 50 random factual statements expressed in a simple sentence, containing only one independent clause. We prompt ChatGPT to produce a list of fact statements that are expressed in a simple sentence. We take 50 of these sentences as our baseline, with hand-filtering to remove any strange or duplicate sentences. We then prompt ChatGPT with these baselines and for each baseline, ask it to give a topicalizaed, clefted, vp-topicalized, and passivized form. Again, a final step involves meticulously going through the transformations to ensure accuracy. See Appendix C for details about the prompts. The result is a dataset containing 50 sentences in their base form. For each base form, a passivized, clefted, topicalized, and vp-topicalized form makes up the complete dataset. Refer to Table 1 for an example of an entry in the dataset, and refer to Appendix A for all the baseline sentences in the dataset (their transformations follow naturally).

4.1.2 Wikitext Dataset

The WikiText dataset (Merity et al., 2016) is a collection of over 100 million tokens taken from "good," i.e., featured articles in Wikipedia. Several earlier papers on influence functions have chosen to use this source, and so we have followed suit.

Influence functions are rather anomalous with respect to language modeling experiments. The language model (GPT2, in our case) is pre-trained on a large dataset D, but then it must also be fine-tuned on a smaller dataset with respect to the same language modelling objective. The influence calculations then determine how influential a certain training instance in the fine-tune dataset is on the generation of a query. This fine-tuned set exists only so that influence will not need to be computed on the entire pre-training dataset, which is massive.

We use the training partition of wikitext-2-raw-v1 as the basis of our fine-tuning set. Into this, we have inserted grammatically transformed sentences from the Grammatical Transformation dataset that are semantically unrelated to the wikitext that they are embedded in.

4.1.3 Made-Up Entity Dataset

But because the pre-trained model may have seen some version of the same data, it does make sense to have another dataset where we rename all entities that appear as subjects in the corresponding, untransformed baseline sentences (the transformations then typically change which grammatical function that entity will have) with completely made-up entities. When we use these renamed, baseline sentences as queries during influence calculations, we can then reasonably be assured that the influence will have come mainly from the correspondingly renamed transformation in the fine-tuned set.

Table 2 shows an entry in the Made-Up Entity dataset, and Appendix A shows all the made-up entities.

4.2 Calculating Sentence Similarity

In directly calculating sentence similarity with sBERT vectors, we simply compute the cosine of the sBERT encoding of a baseline sentence with that of each of its transformations in the Grammatical Transformation Dataset in turn. We used the sentence transformer all-mpnet-base-v2 described in Section 2.1.

When calculating sentence similarity with influence functions, we assume that sentences that are more similar will be more influential. Our made-up entity dataset has been concocted with nonsense names so that the transformed sentence that was inserted into the fine-tuning text will, in spite of its transformation, be the most semantically similar. The influence score will then correspond to how similar they are.

Note that due to the symmetry built into the definition of influence functions, we do not need to explicitly symmetrically close our definition of similarity here.

To support batched calculations, all of our added entries are padded to 20 tokens, long enough to cover the longest transformed sentence in our dataset. With this setup, we can obtain the influence of each transformation on generating its own baseline variant.

4.3 Calculating Entity Invariance

When using sBERT to calculate entity invariance, we calculate:

$$\frac{(e \cdot t)}{|e||t|} - \frac{(e \cdot b)}{|e||b|}$$

where e, t and b are the sBERT vectors for the entity, transformed sentence and untransformed baseline, respectively. Note that this calculation avails itself of sBERT's indifference to the semantic type of its input.

We do this for each transformed sentence, for each entry in the Grammatical Transformation dataset.

With influence functions, we again assume that the congruence or salience of an entity to a particular text will be reflected by a greater influence. We again avail ourselves of influence's indifference to the semantic type of the query, which can be as simple as a referring expression. In our experiments, the entity in question will always be the subject of the untransformed baseline sentence. We subtract the influence of the entity on the baseline from the influence of the entity on the transformed sentence. Padding is the same as with sentence similarity. Figure 1 presents an example of both tasks under both metrics.

5 Results and Findings

Let us first begin by noting that, across both tasks and all syntactic transformations, there is a tight, linear correspondence between sBERT vector cosines and influence scores. Their Pearson correlation is 0.9326, with a p-value of $2.62 \times 10.^{-178}$

As for the specific grammatical transformations, the five rows shown in the tables in this section were chosen because they represent overall trends; the full results for all 50 sentences can be found in Appendix B. In addition, influence scores were scaled with arctan, compressing the range to $-\pi/2$ to $\pi/2$.

Table 3 contains sentence similarity scores using sBERT cosines. For the sentence similarity task, sBERT tends to encode the passivized forms of sentences most similarly to their corresponding baseline sentences. Table 4 contains sentence similarity scores using influence functions. In stark contrast to the sBERT results, influence finds both topicalizations to be most similar to their baselines, whereas passivization is the least similar. In both tables, we can see that the scores are near the top of their respective scales.

Passivization	Clefting	Topicalization	VP-Topicalization
0.9325544834	0.8652806878	0.8628834486	0.90064466
0.9408032894	0.9085036516	0.883110702	0.8844070435
0.9199316502	0.8648024201	0.8657934666	0.8941929936
0.9520395398	0.9430727363	0.9146342278	0.9339743257
0.9660890102	0.8314833641	0.8642077446	0.9086657166

Table 3: Scores of the Sentence Similarity task between the baseline and each of the different transformations using sBERT cosine similarities. Each row corresponds to one row in the Grammatical Transformation dataset, and each column to a grammatical transformation. Note for this and all tables using this color pattern, white represents the smallest value, and dark green is the largest. Rows are independently colour mapped.

Passivization	Clefting	Topicalization	VP-topicalization
1.570795547	1.570795942	1.570796084	1.570796024
1.570796187	1.570796237	1.570796266	1.570796248
1.57079604	1.570796097	1.57079621	1.570796157
1.570795623	1.570796153	1.570796207	1.570796074
1.570793101	1.570796037	1.570796129	1.570796194

Table 4: Scores of the Sentence Similarity task between the baseline and each of the different transformations using influence functions. The scores have been normalized using arctangents.

Table 5 shows the entity invariance scores using sBERT cosines. For this task, sBERT vectors are most invariant to passivization relative to their encodings of the respective baseline sentence, whereas clefting exhibits the most variance. Table 6 shows the entity invariance scores using influence functions. For this particular combination, it is more difficult to spot any sort of trend or preference for one transformation over the others. Both of these scores are difference calculations. In the case of sBERT, the differences are closely range-bound around zero, meaning that the effect of using any grammatical transformation was minimal. In the case of influence functions, the prominence of values near $-\pi/2$ shows that all of the grammatical transformations we experimented with resulted in a suppression of influence scores relative to the baseline subject.

Passivization	Clefting	Topicalization	VP-topicalization
-0.09655714035	-0.1692547202	-0.04888242483	-0.08329671621
0.03376698494	-0.01508197188	0.02062654495	-0.01081442833
-0.09354573488	-0.1332816482	-0.1059363484	-0.1363123655
-0.02574926615	-0.05807337165	-0.0239841342	-0.07041674852
-0.02594101429	-0.09438753128	-0.04366868734	-0.07553547621

Table 5: Scores of the Entity Invariance task between the subject of the baseline sentence and each of that sentence's different transformations using sBERT cosines.

Passivization	Clefting	Topicalization	VP-topicalization
-1.570795954	-1.570796167	-1.570795853	-1.570795884
-1.570796289	-1.570796284	-1.570796288	-1.570796289
-1.570796251	-1.570796245	-1.570796264	-1.570796272
-1.570796283	-1.570796278	-1.570796279	-1.570796278
-1.570796223	-1.570796245	-1.570796238	-1.57079623

Table 6: Scores of the Entity Invariance task between the baseline subject relative to each transformation using influence functions. The scores have been normalized using arctangents.

5.1 Significance of Grammatical Transformations

It is also possible to examine differences in the effects of the four grammatical transformations that we selected. The distributions of the various scores across tasks, both jointly and severally, fail Levene's test of homoscedasticity, so a repeated-measures Friedman's test is the appropriate way to test for significant differences among their medians. Its null hypothesis is that there is no significant difference among the four transformations, which would imply (but not prove) a degree of resilience in the chosen semantic representation. As shown in Table 7, the choice of grammatical transformation is significant in the direct sentence similarity task, regardless of method, but is significant for entity invariance only with sBERT cosines, not with influence functions. Note that the magnitudes of the p-values are at opposite poles, so this is a matter of kind, not degree. Table 8 shows the respective test statistics with their effect sizes. The three significant effect sizes are all considered large, because they are greater than 0.5.

For the settings found to be statistically significant, we present a ranking of transformation preference (higher scores are more preferred) in Table 9. This confirms that for the task of sentence similarity, influence finds passivization to produce the least similarity, and therefore the most difference in meaning, while sBERT finds passivization to be most similar. In fact, while they have similar p-values and test statistics to those for sBERT vector cosines, their ranking of grammatical transformations is the exact opposite.

For entity invariance, on the other hand, sBERT once again finds passivization to best preserve it, although clefting preserves it the least. In both tasks, we are of course not testing whether passivization influences meaning, but rather, given that passivization is thought to be meaning-preserving, whether sBERT cosines and influence functions perform as we want them to.

	Sentence Similarity	Entity Invariance
sBERT	2.32×10^{-15}	3.97×10^{-7}
Influence	1.69×10^{-15}	0.983

Table 7: p-values of Friedman's test for different experimental settings.

	Sentence Similarity	Entity Invariance
sBERT	71.23 / 1.1936	32.57 / 0.807
Influence	71.88 / 1.199	0.17 / *0.058

Table 8: Test statistics (χ^2) / effect sizes (ϕ) of Friedman's test for different experimental settings (the lower-right effect size is hypothetical, as no significance has been demonstrated).

5.2 Effect of Concocted Names

As shown in Table 10, the effect of concocting the names of the fixed entities magnifies the effect of changing the grammatical transformation in the entity invariance task to the point that it becomes statistically significant, and of moderate, almost large size.

6 Discussion

That influence functions might demonstrate any resilience to syntactic transformations is indeed interesting, because: (1) sBERT vectors do not (nor does any other vector-based representational scheme that we are aware of), in spite of how amazingly well they work as semantic representations, and (2) it means that influence functions bring us that much closer to being able to truly work with the meanings of sentences rather than more superficial aspects of their syntactic realizations. Nevertheless, this resilience has only been seen in our examination of something more subtle, where we look not at differences in meaning, but differences in influence scores relative to a fixed entity, and thus arguably differences in differences in meaning. Were it not for entity invariance, in fact, one might wonder why influence scores bothered to exist, given their strong Pearson correlations to sBERT-vector cosines and fickleness with respect to syntactic transformations in more direct comparisons of sentence meaning.

The results on the Made-Up Entity dataset suggest that at least some of the resilience of influence functions is due to their ability to draw upon the meanings of the pre-trained data or the syntactic variety of their expression, or both, in order to see through the effects of a syntactic transformation. In typical LLM fashion, however, the patterns learned by the language model in relation to this are not sufficiently robust or principled to withstand, for example, an innocuous change in the semantic arguments. And so an innovation that was designed to isolate the effects of the query around the transformed sentence in fact hurt performance.

6.1 Limitations

We cannot flatly claim that influence functions are a better alternative to sBERT vectors, in part because of the adverse effects of consistently changing names. There are other limitations, too, the chief of which is that sBERT uses an encoder-style model which contains bi-directional context, while the Anthropic code base and paper for influence functions is focused around GPT2 and other decoder models that only see previous tokens in its history. So it is impossible to determine the extent to which the entity invariance we saw with influence functions is due to the underlying decoder architecture without rewriting that code base. What we can already affirm is that this difference in architecture was not enough for influence scores to fall out of lock step with sBERT cosines in the Pearson correlation test that we conducted.

Another limitation is our choice of a small number of grammatical transformations for experimentation. The results presented in Table 9 naturally single out passivization from the other transformations, and indeed passivization is special. It is the only transformation among the four that we selected to unequivocally constitute A-movement, and the only one that rotates the grammatical function assignment around the arguments of the baseline sentence. It is also the only one of the four that has overt morphological reflexes, although both clefting and VP-clefting would also cause the LLM's tokenizer to change the length of the input. One might also argue that certain of these four transformations are easier to withstand or easier to predict the consequences of, using the measurement tools at our disposal, either because of the structural complexity of the transformation in terms of a chosen syntactic representation, or because of a variance in their relative frequencies in the pretraining corpus. We would, at the same time, like to expand the experimental list of transformations, while better balancing these other effects, but these two purposes work against each other.

	Passivization	Clefting	Topicalization	VP-Topic
Influence on Sentence Similarity	1.570795571	1.570795892	1.570796019	1.570796057
sBERT on Sentence Similarity	0.937302351	0.9078437984	0.8857396245	0.8987811208
sBERT on Entity Invariance	-0.05444133282	-0.08711430431	-0.05307358504	-0.07616019249

Table 9: Medians of the scores on the Grammatical Transformation dataset for each transformation under statistically significant conditions, ranked by colour.

	p-values	Test Statistics	Effect Size
Sentence Similarity	3.79×10^{-14}	65.568	1.145
Entity Invariance	0.008	11.712	0.484

Table 10: p-values, test statistics (χ^2), and effect sizes (ϕ) for different tasks with the Made-Up Entity dataset (influence functions only).

7 Conclusion

Along with neural language models has come increasing concern over transparency and explainability. Influence functions are one example of an attempt to understand or interpret language models. There is some evidence, as shown in this paper, that influence functions are good for more than assigning blame for faulty output. They correlate well with sentence-similarity scores.

Using entity invariance over grammatical transformations, we have been able to distinguish the two, however. While sentence embeddings are not resilient to syntactic transformations in any of our experimental settings, in certain conditions, influence functions are. This is important, because meaning representations should be invariant to meaning-preserving transformations.

It will be important to repeat this experiment after reworking either the Anthropic code base or sBERT so that they can run on the same kind of model. It will also be important to expand and better control the list of syntactic transformations.

References

Lobke Aelbrecht and Liliane Haegeman. 2012. Vp-ellipsis is not licensed by vp-topicalization. *Linguistic Inquiry*, 43(4):591–614.

Juhan Bae, Nathan Ng, Alston Lo, Marzyeh Ghassemi, and Roger Grosse. 2024. If influence functions are the answer, then what is the question? In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS ’22, Red Hook, NY, USA. Curran Associates Inc.

Samyadeep Basu, Phillip Pope, and Soheil Feizi. 2021. Influence functions in deep learning are fragile. In *ICLR*. OpenReview.net.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. The MIT Press, Cambridge.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Thomas George, César Laurent, Xavier Bouthillier, Nicolas Ballas, and Pascal Vincent. 2018. Fast approximate natural gradient descent in a kronecker-factored eigenbasis. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS’18, page 9573–9583, Red Hook, NY, USA. Curran Associates Inc.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamilė Lukošiūtė, Karina Nguyen, Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman. 2023. Studying large language model generalization with influence functions. FAR.ai Alignment Workshop 2023.

Frank R. Hampel. 1974. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML’17, page 1885–1894. JMLR.org.

Knud Lambrecht. 2001. A framework for the analysis of cleft constructions. *Linguistics*, 39(3):463.

Zhe Li, Wei Zhao, Yige Li, and Jun Sun. 2024. Do influence functions work on large language models? *Preprint*, arXiv:2409.19998.

James Martens. 2020. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76.

James Martens and Roger Grosse. 2015. Optimizing neural networks with kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML’15, page 2408–2417. JMLR.org.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ken Shi and Gerald Penn. 2025. Semantic masking in a needle-in-a-haystack test for evaluating large language model long-text capabilities. In *Proceedings of the Writing Aids at the Crossroads of AI, Cognitive Science and NLP WR-AI-CogS Workshop at the 31st International Conference on Computational Linguistics*, Abu Dhabi, UAE.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: masked and permuted pre-training for language understanding. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. *XLNet: generalized autoregressive pretraining for language understanding*. Curran Associates Inc., Red Hook, NY, USA.

Rui Zhang and Shihua Zhang. 2022. Rethinking influence functions of neural networks in the over-parameterized regime. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):9082–9090.

A Full Datasets

Table 11 contains all the concocted entities that, after replacing the subjects in the Grammatical Transformation dataset, form the Made-Up Entity dataset described in 4.1.3.

Table 12 contains all the baseline sentences in the Grammatical Transformation dataset described in Section 4.1.1. The transformations are omitted for brevity but follow directly from the baselines.

B Full Result Tables

Table 13 contains the full results of the sentence similarity task on both metrics. Note that there are 50 rows of data, each corresponding to an entry in the Grammatical Transformation Dataset. Colors are mapped such that the smallest is white, largest is dark green and intermediate values are gradated uniformly, and in addition, each row is done independently.

Table 14 contains the full results of the entity invariance task on both metrics.

Table 15 contains the full results for the test of influence on the Made-Up Entity dataset on both tasks, detailed in Section 4.1.3.

C Prompts

Prompt to generate the baseline fact sentences: *Please provide me a list of factual statements like "Mozart composed symphonies" that follow the simple sentence structure.*

Given the list of baseline sentences, the prompt to generate a transformation: *I will provide you a list of sentences. You are to take each sentence and topicalize it. For example, if given "John liked Mary." you are to return "Mary, John liked."*. The same prompt can be adapted for the other transformations.

Made-up Entities in the Made-Up Entity Dataset

Kolpytimia	Fervan	Phran	Zorvik	Ivoren
Jymilopy	Galros	Quirin	Reilktyia	Jexar
Fulkingra	Hivian	Raxen	Avaron	Kynor
Liuntmat	Ivren	Salven	Brenix	Larven
Kolparop	Jovik	Torvin	Cyrin	Morden
Funmilip	Kelrin	Uvorn	Dralin	Nexor
Belrix	Laxor	Vexan	Elvir	Shadrin
Cevran	Merin	Wavric	Fixon	Fullinma
Darvon	Novin	Xalden	Gravin	Dilkop
Emlian	Orvex	Yavren	Haldor	Imnity

Table 11: All entities in our Made-Up Entity Dataset. These replace the subjects of the Grammatical Transformation dataset to form the new dataset

Baseline Sentences in the Grammatical Transformation Dataset

Albert Einstein developed the theory of relativity.	Armstrong landed on the moon.
Isaac Newton formulated the laws of motion.	Fleming discovered penicillin.
Leonardo da Vinci painted the Mona Lisa.	Darwin explained evolution.
William Shakespeare wrote Hamlet.	Jobs founded Apple.
Marie Curie discovered radium.	Beethoven composed Fur Elise.
J.K. Rowling wrote the Harry Potter series.	Hillary climbed Everest.
Vincent van Gogh painted The Starry Night.	Pasteur developed vaccines.
Nikola Tesla invented the alternating current (AC) motor.	Galileo built telescopes.
Georgy Zhukov led the defense of Stalingrad.	Ford revolutionized manufacturing.
Alexander Fleming discovered penicillin.	Orwell wrote 1984.
Michelangelo sculpted David.	Picasso painted Guernica.
Charles Darwin developed the theory of evolution.	Edison patented the light bulb.
Thomas Edison invented the electric light bulb.	Mandela fought apartheid.
Beethoven composed Symphony No. 5.	Turing cracked the Enigma code.
Alexander Graham Bell invented the telephone.	Pythagoras discovered the Pythagorean theorem.
Mozart composed The Magic Flute.	Hitchcock directed Psycho.
Leonardo DiCaprio played the role of Jay Gatsby.	Mozart composed Don Giovanni.
Columbus discovered America.	Washington led the Continental Army.
The Wright brothers invented the airplane.	Napoleon invaded Russia.
Alexander conquered Persia.	Franklin invented the lightning rod.
Marie Curie studied radioactivity.	Curie discovered polonium.
Tesla designed alternating current systems.	Kepler described planetary motion.
The Romans built aqueducts.	Gagarin orbited Earth.
Magellan circumnavigated the globe.	Caesar crossed the Rubicon.
Gutenberg invented the printing press.	Chopin composed nocturnes.

Table 12: All 50 baseline sentences used in the Grammatical Transformation Dataset. Not included for brevity are the corresponding grammatical transformations but they all follow naturally to make up the full dataset.

Sentence Similarity via Influence				Sentence Similarity via Sentence-BERT			
Passivization	**Clefting**	**Topicalization**	**VP-Topicalization**	**Passivization**	**Clefting**	**Topicalization**	**VP-Topicalization**
1281673.25	2596264	4117991.25	3298893.25	0.9325544834	0.8652806878	0.8628834486	0.90064466
5032188.5	7685314	6797391	6382516	0.9408032894	0.9085036516	0.883110702	0.8844070435
7136764.5	11084613	16475059	12699568	0.9306652546	0.8681627512	0.869109869	0.8732848167
3492748	4352391	8556159	5876025	0.9199316502	0.8648024201	0.8657934666	0.8941929936
1532958	2286936.75	1962189.875	2141318.75	0.9520395398	0.9430727363	0.9146342278	0.9339743257
4620397	13047812	8191310.5	20385430	0.9660890102	0.8314833641	0.8642077446	0.9086657166
1275383.625	2828912.75	4700864.5	6202506.5	0.9333539009	0.9185542464	0.9130138755	0.8701131344
3085148.25	2570608.75	3505769.5	5189386.5	0.9549874067	0.9168089628	0.9037501812	0.9442888498
5773955.5	7070393.5	10300728	12519818	0.9459875226	0.9487894773	0.9275122881	0.8671823144
4353710.5	3066081.75	5515658	3065397.75	0.9545772076	0.9427666068	0.9304442406	0.9420560598
1201822.25	762331.3125	6281364	5113827	0.9067315459	0.9071839452	0.8697237372	0.902159512
6461203	5040498.5	7943203.5	8796788	0.9197968245	0.8237189054	0.8344243169	0.8498998284
1900646.625	1747893.375	3532316.25	5013893	0.9284735918	0.84999156	0.8362667561	0.9234173894
3811932	5775193	5407366.5	8614384	0.9404629469	0.940613687	0.9342517853	0.8057485819
8396329	13875857	38902156	-336879.2813	0.9341417551	0.8465870023	0.7667613029	0.8968443274
2837912.5	2746652.75	3477327.5	5668458	0.9407648444	0.7814874649	0.8770526648	0.8991389275
2148719.75	2310032.5	3563150	7584823.5	0.9522520304	0.9452135563	0.8985278606	0.836519599
1163579.875	1002357.438	2497049.5	1499065.875	0.9046645164	0.8813423514	0.7317293882	0.8887551427
1367728.375	614765.3125	2399663.5	2348435.25	0.9161099195	0.8231762052	0.8400527835	0.8983151913
1972371.625	5103495.5	3946486.5	6394397.5	0.9530593753	0.918993175	0.8561660051	0.9100579023
712751.5	1288459.25	2425012.75	3654488	0.9562042356	0.9505699277	0.9037286043	0.8873476982
3226312.25	12242624	3931645.75	13437938	0.9615622759	0.9444450736	0.9279776812	0.8876610994
1420691.625	5752419	8330069	3952169	0.9537521601	0.9556134939	0.9316477776	0.8822870851
309985.2188	3447946	5053616	7553659	0.9246538281	0.8528832197	0.856222868	0.9120983481
800681	538536.625	2079399.75	2897636.25	0.9088691473	0.8832570314	0.8298295736	0.8844642043
91632.98438	1477246.75	1883422.875	2362679.5	0.8620303273	0.7549761534	0.7469062209	0.8201477528
2736904.5	441928.3438	763860.5	1111945.375	0.9550385475	0.9451751709	0.9499857426	0.9289374352
317121.375	905179.6875	2216923.25	1574170.75	0.8973581791	0.8525787592	0.8231647611	0.8734014034
1058726.875	2258758.75	3243198.25	4167219.75	0.9199647903	0.8848507404	0.8137908578	0.903968513
-1187882.375	6206952.5	1382713.875	2774814.75	0.9419152141	0.9371224046	0.8946403861	0.9033447504
1384578.5	11605822	1470714.5	12923023	0.8947380781	0.8980829716	0.8914081454	0.878882587
398852.7188	877115.1875	2340028.25	1418243.75	0.9431471229	0.9407480955	0.927508533	0.8732652068
512260.9688	124985.6953	1507217.375	3009961.5	0.9419971704	0.9371962547	0.9451744556	0.9277190566
927574	-569455.8125	890655.3125	3681423	0.9550658464	0.9444385767	0.9141231775	0.9183707237
733503.875	1276804.125	3643654	613235	0.9289010763	0.8879346251	0.8347960711	0.8943598866
2617136.25	1795073.5	2855236	7847653	0.9220842123	0.915694356	0.8794906735	0.8984233141
636359.25	126831.7031	1154313.75	4479112.5	0.9173202515	0.9273391962	0.895643115	0.9292954206
688857.4375	870001.75	1613224.25	2188606.25	0.8809921145	0.8822927475	0.8651847839	0.8980981708
897886.75	2438285.5	3755775.25	5557944	0.9443784356	0.8753024936	0.8788477182	0.900886178
941638.5625	1710459.5	3206986.75	2542614.5	0.9473628402	0.9563817978	0.9277408123	0.9360141158
30088.2207	3036802.75	1362985.5	3273471.25	0.8767876625	0.8422478437	0.918872118	0.8392100334
-774782.5625	4324408	3250920.75	7688956.5	0.9330461621	0.922550559	0.888368547	0.9031774402
1543971.625	1883542.375	2346876.5	2069620.625	0.9551422596	0.9336919785	0.9073114991	0.9046003222
2149576.5	1670727.625	4511604.5	2736040.75	0.9525103569	0.8796239495	0.8039262891	0.8643612266
-66643.16406	208990.6719	760295.8125	2674516.25	0.9314661622	0.906768024	0.925755322	0.924367547
1650114.375	3372535	1811241.375	3720192	0.9604322314	0.9507023096	0.9522266388	0.9335971475
2179720.75	94931.34375	1805788.5	2851061.25	0.9632445574	0.9436131716	0.9506351948	0.9086754918
908247.5	2219431.75	6535197.5	2098055.5	0.9507032633	0.9038532972	0.8263111115	0.9095230699
-470488.875	688643.0625	2670438.5	3444965.75	0.9334220886	0.9068481922	0.8767338395	0.9065231085
-1013974.75	6141339	2260686	7763943	0.9230386019	0.9325930476	0.9073643088	0.8753144145

Table 13: Full results of Sentence Similarity for both metrics on the entire Grammatical Transformation Dataset

Entity invariance via Influence				Entity invariance via Sentence-BERT			
Passivization	Clefting	Topicalization	VP-topicalization	Passivization	Clefting	Topicalization	VP-topicalization
-2685623	-6254405	-2108614	-2258712	-0.07981181145	-0.1866739392	-0.1492462158	-0.07941681147
-26642611	-23641762	-25821433	-26574901	-0.06372123957	-0.1190310717	-0.112989068	-0.0492888093
-13220497	-12153932	-15876012	-18372530	-0.08093649149	-0.1657860875	-0.06755018234	-0.04545301199
-23089960.75	-20474648.5	-20957768	-20638588.5	-0.1250346899	-0.1121886373	-0.05282765627	-0.1104551554
-9629271.5	-12170401.5	-11296740.5	-10278239.5	-0.004707098007	-0.06935936213	-0.05996596813	0.01575297117
-27716518	-24282692	-25155550	-24522918	-0.02630108595	-0.1617150307	-0.1105882525	-0.1176011562
-25609095	-25669994	-24993926	-25406562	-0.0555267334	-0.07726699114	-0.0490463376	-0.05264532566
-14933946	-12275290	-16347746	-16882840	-0.08301895857	-0.100716114	-0.07386052608	-0.08472329378
-15794400	-8453728	-11933606	-11977778	0.003785192966	-0.02281010151	-0.06004858017	-0.1017688513
-30020002	-19234332	-26009050	-21754662	-0.04582571983	-0.06943738461	-0.06015014648	-0.02689957619
-48643260	-52031866	-48561260	-46328487	-0.1114506721	-0.09153693914	-0.05393457413	-0.1327273846
-13833647	-9833368	-13587115	-10178348	-0.06523412466	-0.1308091283	-0.08723050356	-0.06536006927
-26825080	-25047562	-26189696	-25613158	-0.09480243921	-0.1450120807	-0.1342134476	-0.05758196115
-1135976.25	-475462.5	-2344986.125	-1476259.547	-0.0925809741	-0.1185005307	-0.05976593494	-0.06406724453
-39184466	-38029180	-31495636	-36814442	-0.04791623354	-0.1590764523	-0.1918034554	-0.08462017775
-1247697.25	-1582920.375	-2108257.125	-1594213.875	-0.09655714035	-0.1692547202	-0.04888242483	-0.08329671621
-5411460	-6072064	-5347273	-5201701	0.03376698494	-0.01508197188	0.02062654495	-0.01081442833
-4791516.125	-3093396.25	-3368433.25	-3402382.5	-0.09742739797	-0.08908066154	-0.006914794445	-0.09745392203
-12588536	-28372232	-34465044	-39599843	-0.09354573488	-0.1332816482	-0.1059363484	-0.1363123655
-5319618.375	-4277966.313	-4288703.438	-5360857.469	-0.02574926615	-0.05807337165	-0.0239841342	-0.07041674852
-7629488	-11472516.5	-10119478.5	-9223493	-0.01981073618	-0.03779411316	-0.05990833044	-0.03124922514
-1823057.594	-1784169.672	-1747495.781	-1909726.992	-0.03912311792	-0.07392579317	0.03000319004	-0.06897968054
-31943223	-28038618	-34662588	-32163390	-0.05335593224	-0.03372785449	-0.01171341538	-0.02090236545
-1104358.875	-2584393.844	-3484716.25	-1595341.125	-0.02594101429	-0.09438753128	-0.04366868734	-0.07553547621
-3454303.375	-1707085.625	-1072567.125	-1806547.125	-0.06934568286	-0.07635483146	-0.07115519047	-0.1302825809
2332928.625	2165193.875	2180960.313	2484504.125	-0.08078327775	-0.1065143049	-0.1084765792	-0.1459647715
-12339632.25	-13939776.63	-16306948.5	-15179150.88	-0.03378689289	-0.02639275789	0.001132577658	0.02219408751
-2894744.125	-3501600.211	-3639278.414	-3362675.984	-0.04256004095	-0.02782595158	0.06826972961	-0.003503620625
-198664.1445	-228390.333	-185710.7344	-199948.0586	-0.06719768047	-0.1265891194	-0.01654732227	-0.07524868846
-414397.375	-730582.7031	-678280.2813	-781506.2969	-0.09697979689	-0.08204746246	-0.06582641602	-0.05791759491
-283525.1289	-284416.8887	-261344.5234	-260542.9336	-0.09624645114	-0.08741539717	-0.04021796584	-0.07799932361
-8322824	-7754106	-6927294	-7234931	-0.03954720497	-0.04791337252	-0.01593309641	-0.1249685585
-38572803	-24507834	-34113582	-31726072	-0.03267228603	-0.04540675879	-0.02843618393	-0.04879248142
-18167.93164	-18284.80469	-22466.16797	-15712.23633	-0.02734774351	-0.06580168009	-0.05387979746	-0.1557758152
-446892.125	-483367.3125	-479492.9375	-1020807	-0.07068240643	-0.1144337654	-0.09733355045	-0.07694244385
-3250265.25	-4003595.563	-3033394	-3140831.75	-0.1278484464	-0.1360321045	-0.0533195138	-0.09156519175
-1908188.188	-831025.9375	-792524	-1234735.563	-0.1284969449	-0.1011826992	-0.09552234411	-0.04509288073
-1874450.125	-1568216	-1688667.938	-1822333.188	-0.1489322186	-0.1448811293	-0.06250846386	-0.1114014387
-4128723.789	-2868800.25	-5321833.625	-1356737.25	-0.04154163599	-0.07667589188	-0.0476590395	-0.07678490877
-29504152	-32576832	-30273304	-28288826	-0.03211379051	-0.04234272242	0.02607136965	-0.01085174084
26074.9375	-451694.0859	-220842.4688	-318559.5	-0.03710752726	-0.1560547352	-0.09752297401	-0.08025348186
-1946600.25	-2015420.75	-2433695.125	-1350453.25	-0.0997890234	-0.02228420973	0.01836383343	-0.08943325281
-718451.7813	-944356.6641	-945010.3594	-978230.1328	0.02094578743	-0.02048495412	0.01983216405	-0.1224358678
-29409283.88	-30200487.56	-29657554.56	-29677078.81	0.01468878984	-0.08090877533	-0.007811784744	-0.05628025532
-1109870.063	-1860999.219	-1730642.281	-2111163.311	-0.04685598612	-0.05257755518	-0.0422347784	-0.04875138402
-2075367	-2253507.375	-2368550.75	-2355812.188	-0.04864227772	-0.0438978672	-0.04264587164	-0.02939426899
-326502	-2257342.25	-2389740.875	-1023074	-0.06806963682	-0.08834481239	-0.0005748867989	-0.1109085083
-17051838.84	-16849220.94	-15265631	-16327568.88	-0.03762674332	-0.08681321144	-0.09053331614	-0.1175132394
-2649977.906	-2593891.906	-2393907	-2046743.75	-0.02224761248	-0.04226249456	-0.0484764576	-0.06495755911
-26235076.38	-26820000.16	-26304973.25	-26184891	-0.097905159	-0.08832764626	-0.02254664898	-0.1072673202

Table 14: Full results of Entity invariance for both metrics on the entire Grammatical Transformation Dataset

Sentence Similarity via Influence				Entity Invariance via Influence			
Passivization	Clefting	Topicalization	VP-topicalization	Passivization	Clefting	Topicalization	VP-topicalization
20699412	18898032	18216750	14871124	-27748200	-21957264	-19352244	-16531028
16107982	16881070	22166466	18357252	-25357752	-31761387	-25464259	-24658304
8983601	9679158	13356021	15754497	-15623932.5	-15539081	-10364748	-19133420
6311177	31551756	15346468	33377176	-32253152	-34866796	-30644562	-31838056
-4400801	3247743.25	8152989	-1417421.375	-37055640	-34900592	-44443604	-41120728
7341449.5	4947732.5	10200429	8779436	-10611611	-9151762	-10131612	-9676926
4204987.5	3982068	3914922.75	14536122	-2995158.75	-2896915.563	-2880270.25	-3905569.219
39287480	39517636	41943084	43717972	-23887976.25	-24590565.5	-24768332.5	-24817787.5
5499680.5	5292868	9293386	14166252	-6988639.25	-8419173.469	-8611006.188	-8578112
795575.6875	721371.75	2425898	1754051.875	-16320768	-17970983	-18785290	-17531584
1111503.125	2716567.25	1668867.125	5971097	-46875189.25	-48718234.09	-48014607.44	-47222828.75
3494235.5	5168803	5021732.5	6507909	-1306780	604563.5	-1818692	-198710
2854848.5	2777719.5	3898743.25	5461473	-44338464	-56349592	-50853727	-52246354
1941477.5	9786865	5676007.5	9990937	2108757.125	2131060.875	1837815.625	1714763
806739.75	1078171	3864495.25	1805136.25	-3313877	-5654521	-6208624	-6319545
1971969.5	2315744.5	3322627	4030830	609971	-116435	82594.09375	-293907.0938
3113887.25	3309663.25	5848674	7088668	-61137592.38	-62781103.19	-61198772.5	-61969797.77
904815.0625	4273071.5	1094813.375	3007366	-4451510.205	-4695470.813	-4905691.969	-4977761.5
476680.4375	180478.6094	677784.3125	1363827.125	-8516562.813	-8411518.031	-8619159.438	-8473698.125
-1440849.125	6980230.5	5300396	4356878.5	-11607418	-14793079	-14087506.5	-15849524.5
1288198.375	3206311.25	5574628	6867961	3482046.375	3396289.5	2969881.625	2794664.625
3750752	9526400	4090863.75	12548599	1851081.109	1852303.016	1861345.625	1741095.734
1479134.5	2008157.5	1753084.5	1594822.125	-2386763.75	-2982937.656	-2696552.434	-2792250.531
622985.6875	2346483.75	6988319	7975886	-18440996.5	-23206848	-19082092.5	-22463687
537622.6875	960032	2324116.75	3750187.75	-1748337	-3400219.875	-2762614.938	-3739159.875
1378174.875	1122773.25	2615694	3123351.5	-34615689.75	-35094075.25	-34674898.25	-35754053.25
347643.125	594932.75	1796586.375	2670808.75	-11495633.88	-11424431.88	-11695386.09	-11622510.38
305673.8125	661324.9375	2004349.625	1520810.75	-4411322.625	-3197441.75	-4706700.25	-4590533
2249547	3402734.5	6386406	4276829.5	-25215807	-29706655	-25352677	-28966407
3405452.25	12879227	5271509	8707653	-11760492	-14378362	-13283870	-10874287
768798.9375	10446396	5316533.5	12891541	-20133013.5	-20565648.5	-21723378	-20852058
9640745	9694372	6758299.5	3705323.75	-16910541	-22972139	-16782637	-18804143
355867.75	1285882.375	1297126.75	1909627.625	344915.875	161834.25	420683.5	-502630.875
1240693.875	2390800	3266505.75	3750562.5	-84545504	-86513072	-95985880	-85100844
126504.9453	689580.875	1197900.25	1589813.875	-2965967.5	-2691341.563	-2771703.875	-2990364.906
10040059	7601738	3605134.75	6452776.5	-12713596	-14162309	-9163776	-13836231.5
574604.875	1275667.75	2521754	3318908.25	-241800.1602	-507822.1719	-1196177.531	-633820.4688
229866.9063	489298.1563	1058893.375	4325071	-1023570.063	-227643	-961018.6875	136663.125
1205245.625	2386297.75	4134517.75	4209640	-8267742.375	-9116313.133	-9054034.375	-8748847.125
4838220.5	5989610	7196361	7528256.5	-170502362	-156098350	-144139452	-155251562
680086.625	7122028.5	3757842	4863109.5	-43254391.25	-44739429.88	-42880126	-40628191.5
6614905.5	8474995	4974632.5	12607139	-21855384	-24289794	-20991455	-21518870
1809379	3166296.5	4690755	2627413.25	-55810420	-60561396	-52134966	-56077972
2263918	830624.4375	4329830.5	10571793	-18316043.25	-16365657	-14891428	-15227425.5
-422677.3125	232356.2344	1123875	2746210.5	1260457.25	-1450561.25	-828111.5	101014
393759.875	1359005.75	1329041.125	2745960.25	2505653.137	2954838.031	2476717.504	3066468.438
1965582.375	4272043.5	3442731	4518502	-10066473	-13644846	-16204924.75	-13169884.5
567223.3125	9596649	3692350	7254443.5	-12858188	-19651287	-17660868.5	-15906087
-179255.6563	-3388808	7529950.5	3677222.75	-21820827.5	-21620684	-21853104	-21528893.75
1040165.625	6428225.5	6093796.5	1911645.75	-34843994.5	-35182859.5	-34070036	-34218104

Table 15: Full results of Influence on both tasks for the Made-Up Entity dataset

Automatic Extraction of Clausal Embedding Based on Large-Scale English Text Data

Iona Carslaw*[1, 2], Sivan Milton*[1, 2], Nicolas Navarre*[1, 2],
Ciyang Qing[2], Wataru Uegaki[2]

[1]School of Informatics, University of Edinburgh
[2]School of Philosophy, Psychology & Language Sciences, University of Edinburgh
{ I.C.A.Carslaw, s.milton, n.s.navarre}@sms.ed.ac.uk {cqing, w.uegaki}@ed.ac.uk

Abstract

For linguists, embedded clauses have been of special interest because of their intricate distribution of syntactic and semantic features. Yet, current research relies on schematically created language examples to investigate these constructions, missing out on statistical information and naturally-occurring examples that can be gained from large language corpora. Thus, we present a methodological approach for detecting and annotating naturally-occurring examples of English embedded clauses in large-scale text data using constituency parsing and a set of parsing heuristics. Our tool has been evaluated on our dataset Golden Embedded Clause Set (GECS), which includes hand-annotated examples of naturally-occurring English embedded clause sentences. Finally, we present a large-scale dataset of naturally-occurring English embedded clauses which we have extracted from the open-source corpus *Dolma* using our extraction tool.

1 Introduction

One of the most popular methods of conducting linguistic research has consisted of handcrafting paradigmatic utterances followed by gathering native speakers' judgements. Yet, it is questionable how much these constructed utterances reflect real-world language use. As a result, plenty of debate has arisen about the legitimacy of paradigmatic utterances as a research tool, with arguments suggesting this particular data collection technique can lead to biased results (Cowart, 1997; Schütze, 2016). Whilst this debate has been happening in linguistics, the advancements of Natural Language Processing (NLP) have led to a significant increase in the amount of freely available language corpora as well as an increase in their size. For example, the open-source dataset *Dolma* consists of 3 trillion English tokens (Soldaini et al., 2023). These datasets provide new opportunities for linguistic research, with the ability to gather statistical data about specific language constructions and naturally-occurring examples beyond handcrafted sentences.

One particular sentence construction that would benefit from such corpus research is that of EMBEDDED CLAUSES. These constructions contain an embedding predicate which selects a clausal complement, as seen in the sentence: *Mary hopes that John likes chocolate*. Here, the predicate *hopes* embeds the declarative clausal complement *that John likes chocolate*. Alongside DECLARATIVE clausal complements, as in (1a), there are also POLAR INTERROGATIVE clausal complements (1b), ALTERNATIVE INTERROGATIVE clausal complements (1c), and CONSTITUENT INTERROGATIVE clausal complements (1d). Crucially, predicates vary with respect to which clausal complement type they are allowed to embed; consider the difference in grammaticality between *wonder*, which can embed interrogative clausal complements, and *hope*, which cannot embed interrogative clausal complements.[1] In addition, it has been observed that emotive factives, such as *be happy (about)*, take declarative and constituent interrogative complements but not polar and alternative interrogative complements (Abels, 2004; Karttunen, 1977; Sæbø, 2007, a.o.).

(1) a. Mary {*wondered | hoped | was happy } [that John liked chocolate].

b. Mary {wondered | *hoped | *was happy about } [whether John liked chocolate].

c. Mary {wondered | *hoped | *was happy about } [whether John liked chocolate or cake].

d. Mary {wondered | *hoped | was happy

[*] These authors contributed equally and are ordered alphabetically.

[1] These judgements, commonly reported in the literature, are shared by the 3 native British English and Canadian English speakers among the authors.

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 322-332.
Eugene, Oregon, July 18-20, 2025

about } [which chocolate John ate].

Because of this observation that a predicate selects for particular types of embedded clause in fine-grained ways - partly conditioned by the predicate's lexical semantics - there is a debate amongst syntacticians and semanticists about what roles syntax and semantics play within these constructions (Grimshaw, 1979; Uegaki and Sudo, 2019; White, 2021, a.o.). Extrapolating clausal embeddings from large-scale corpora would help to answer such questions, by providing large-scale statistical evidence for how often these embedding predicates appear in natural language use and what clausal complements they select, as well as the ability to look for natural language examples. Thus, the aim of this paper is to create a tool for linguists to extract English sentences containing embedded clauses from large-scale corpora, whilst also providing the following information: (i) the span of the embedded clause, (ii) the lexeme(s) of the embedding predicate, and (iii) the type of the embedded clause.[2]

This task of extracting embedded clauses is by no means trivial. Firstly, the span of the embedded clause in a sentence has to be correctly identified, excluding any element that belongs to the matrix clause. Secondly, there are constructions that superficially resemble embedded clauses, but are in fact not, as they fail to categorise syntactically as *complements* of an embedding predicate or as *clauses*. To see this, consider the following examples:

(2) a. Mary saw a man [that John mentioned].

b. Mary ate [what John cooked].

c. Mary goes to the gym regardless of [whether she is tired or not].

The bracketed clause in (2a) is a RELATIVE CLAUSE and is not a complement of an embedding predicate. In (2b), we have an instance of a FREE RELATIVE, which is considered as primarily a Noun Phrase rather than a clause (Caponigro, 2003; van Riemsdijk, 2006). The bracketed clause in (2c) is an UNCONDITIONAL (Rawlins, 2008), which is a modifier rather than a complement of a matrix predicate. Thirdly, embedded clauses can arise in complex clausal structures such as coordination (3a), which often occurs with ellipsis, nesting (3b), or some combination of both (3c). Consequently, to correctly identify embedded clauses, we need a correct syntactic parse of the sentence, as well as appropriate heuristics to rule out structures such as those in (2) and deal with the structures in (3).

(3) a. Mary knows [that John likes chocolate] and [that Mark does not].

b. Mary knows [that John thinks [that Mark likes chocolate]].

c. Mary knows [that John thinks [that Mark likes chocolate]] and [that Mark does not].

Our paper is structured in the following way: Section 2 describes previous attempts at building a large-scale corpora of English embedded clauses (e.g. MegaAcceptability), and additionally examines existing tools designed to extract sentences from language corpora (e.g. linguistic search engines). Section 3 introduces our hand-annotated dataset of English embedded clauses: Golden Embedded Clause Set (GECS). Section 4 describes our extraction tool that uses constituency representations and parsing heuristics, as well as our tool's performance on GECS. Section 5 presents the large English embedded clause dataset that we have extracted from the open-source dataset *Dolma*. Section 6 suggests future research avenues and 7 concludes our work. Overall, we provide three new contributions:

1. A small-scale dataset (GECS) with fine-grained gold standard annotation of English embedded clauses to be used as a benchmark for this task

2. An extraction tool which can be applied to English language corpora to extract and annotate embedded clauses

3. A large-scale extracted set of English embedded clauses from the language corpus *Dolma* for the linguistic community to use

2 Relevant Work

2.1 MegaAcceptability

The only existing attempt at a large dataset of English embedded clauses is the MegaAcceptability dataset (White and Rawlins, 2016, 2020). White and Rawlins selected a list of 1007 English verbs

[2] Code: https://github.com/navarrenicolas/clause_parser/. Extracted embedded clause dataset available on HuggingFace: https://huggingface.co/datasets/nnavarre/Embedded_Clauses-dolma_v1_6-sample

that are known to select clausal embeddings, and then designed 50 schematic sentences covering a range of syntactic environments in which an embedded clause can occur. They then slotted the 1007 verbs into the 50 schematic sentences to create $\sim 50,000$ entries. Through Amazon MTurk, participants rated the acceptability of the resultant sentences, leading to a large dataset of embedded clause constructions ranked by acceptability, on a 7-point ordinal scale.

Although the MegaAcceptability dataset moves away from the problem of a small set of sentences being used as evidence for linguistic hypotheses, it still utilises non-natural sentences which have been handcrafted. Furthermore, for finite embedded clauses White and Rawlins (2016; 2020) only considered environments without complementisers or with the following complementisers: *that, whether*, and *which*. They also only consider predicates with no prepositions or with the following prepositions: *to* and *about*. They make use of a pre-defined list of verbs which accept clausal complements, which does not account for the full set of embedding verbs nor adjectives and complex predicates which can also accept clausal complements. Therefore, it is unclear if the dataset captures the natural distributions of embedding predicates, embedded clause types, and the types of embedded clauses selected by embedding predicates.

2.2 Linguistic Search Engines

The goal to extract sentences with certain linguistic phenomena from natural language use is not a new concept. There have been several attempts to create *search engines* in which an individual can query annotated natural-language corpora for certain constructions and then be provided with a list of sentences which match the provided query. Prominent tools with this use include the Linguist's Search Engine (Resnik and Elkiss, 2005), SPIKE (Shlain et al., 2020), and the LINDAT/CLARIAH-CZ PML Tree Query (Pajas et al., 2009).

Although these are powerful tools, their query languages are not sufficiently fine-grained to capture the relevant structures of embedded clauses. They rely on annotation of corpora with lemmas, part-of-speech tags, and dependency graph representations. This means that one would need to specify dependency relationships rather than constituency/hierarchical ones to identify the structure of embedded clauses. Such an approach is limiting, as it is difficult to identify clause and predicate spans based on dependency relations or linear structure. There is also less consistency with respect to the relations that identify embedded clauses than with constituency parsers. Moreover, linguistic search engines offer linguists limited flexibility to decide which corpora they want to extract sentences from.

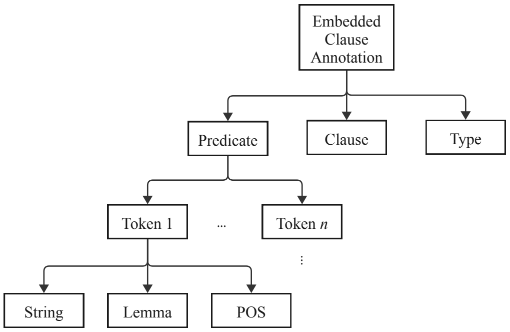


Figure 1: The annotation in GECS for each embedded clause.

3 Golden Embedded Clause Set (GECS)

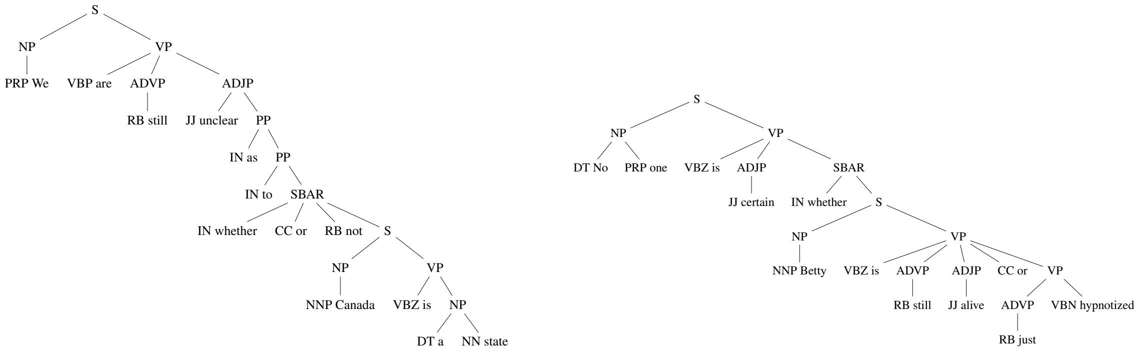
For the novel task of English embedded clause detection in natural language corpora, we created a hand-annotated dataset (GECS) which can serve as a benchmark for evaluation and be used in its own right for a small-scale analysis of embedded clause constructions. In GECS, each embedded clause is annotated with its embedding predicate, clause span, and clause type (see Figure 1). We provide the embedding predicate as a list of the relevant tokens (i.e. ignoring negation words, adjuncts, and any other tokens which may appear between the first embedding predicate token and the clause).[3]

Annotation Procedure To create our naturally-occurring embedded clause dataset, we selected a subset of $866,538$ sentences from *Dolma*[4] (Soldaini et al., 2023). The data was not cleaned so as to accurately test the robustness of the tool. We then parsed the sentences and filtered them to remove any which necessarily did not contain embedded clauses.[5] To extract the set of polar and

[3] To capture the complex constructions of coordinated and nested embedded clauses alluded to in Section 1, we optionally provide a recursive data structure version of GECS which makes the internal clausal structure transparent (see Figure 4 in the Appendix).

[4] Specifically we used the files *cc_en_head-0000*, *cc_en_head-0001*, and *c4-0085*.

[5] We used SBAR from SpaCy's Berkeley Neural Parser (Kitaev et al., 2019; Kitaev and Klein, 2018) as an indicator,



(a) Complex embedding predicate (‘are unclear as to’)

(b) Adjectival predicate (‘is certain’)

Figure 2: Example parses of embedded clause sentences in GECS.

alternative interrogative embedded clauses, we further filtered out sentences that did not contain the words *whether* or *if*. Finally, to filter for constituent interrogative embedded clauses, we only considered sentences with: *who*, *what*, *when*, *where*, *why*, *how*, or *which*. The next stage of hand annotation consisted of one researcher going through the pre-filtered sentences and confirming (i) if there were embedded clauses and (ii) if so, providing the annotation of predicate tokens, clause span, and type. A second researcher then went through the previous researcher's annotations to confirm agreement.

Overall, GECS contains 147 declarative embedded clauses, 138 polar interrogative embedded clauses, 84 alternative interrogative embedded clauses,[6] and 158 constituent interrogative embedded clauses. In addition, we provide a set of 111 adversarial examples verified to not contain any embedded clauses, but do contain misleading structures such as free relatives and relative clauses. These were created by selecting sentences discarded by the annotators in the final stage of GECS' creation.

4 Parser Tool

Though it is possible to define a set of heuristics based on Regular Expressions or dependency relations, preliminary analysis indicates significant disadvantages to such an approach, as were seen with the linguistic search engines from Section 2. For this reason, we opted for representations from constituency syntactic parsers to extrapolate hierarchical structure. A benefit of this choice is that linguistic theory is typically given with respect to constituency trees, and we can therefore implement linguistic facts into extraction heuristics more freely than with other representations. While it is possible that a Dependency Parser could be used to achieve equivalent results, it is not clear what improvements it could offer. We leave this question open to a more thorough exploration in future. With the constituency representation we defined a set of heuristics to perform the following tasks:

1. **Detection**: detecting embedded clause(s) in a sentence
2. **Predicate Identification**: identifying each embedding predicate
3. **Clause Identification**: identifying the span of each embedded clause
4. **Typing**: identifying the type of each embedded clause

The syntactic parser that we use is SpaCy's Berkeley Neural Parser, a constituency parser that has an LSTM and self-attentive architecture (Kitaev and Klein, 2018; Kitaev et al., 2019). Other options are available for constituency parsing; however, we decided upon this parser because it is state-of-the-art for constituency parsing.

The SpaCy constituency parser represents each sentence as an n-ary tree structure with several syntactic categories (e.g. S, VP, NP, SBAR) in parent and child hierarchy. This tree structure is particularly helpful in extracting embedded clauses because we can traverse the parent levels and check for particular child nodes in complement positions. We then defined heuristics based on the structures from the parser to perform the aforementioned tasks of embedded clauses detection, predicate identification, clause identification, and typing.

assuming that this structure in the parsed representation is a necessary condition for an embedded clause.

[6] There are fewer alternative interrogative embedded clauses due to this type being far sparser in the pre-filtered dataset than the other types.

4.1 Methodology

Detection The first heuristic we deemed necessary for detecting embedded clauses is the existence of an SBAR in the parsed representation. This is a syntactic category for a subordinate clause, which is a superset of embedded clauses, but also includes non-embedded clauses like relative clauses. To check if a subordinate clause is an embedded clause, we assume it needs to be dominated by a VP headed by a predicate. While there may be other syntactic categories immediately above the subordinate clause, we are only interested in the first upstream occurrence of one of two syntactic categories: NP or VP. In the case where the label is VP, the sentence has an embedded clause. If the label is NP, then the sentence does not have an embedded clause—likewise if neither of the two are found until the root node of the tree. We use the hierarchical nature of the constituency parser to distinguish embedded clauses from relative clauses and complements of NPs.

To limit the amount of false positives that would be extracted from the dataset we implemented a few heuristics based on the embedding predicate and the subordinating conjunction of the clauses that are detected. First, if the embedding predicate is empty after the part-of-speech filtering or the only predicate token is 'is/be', then the clause is not considered to be an embedded clause. Secondly, we rule out any clauses beginning with certain subordinating conjunction because they are not indicative of an embedded clause. Specifically, we blacklist the following: *after, although, before, despite, to, for, so, though, unless, until, than, because, since, while, as, even if, in order*.

Predicate Identification Having identified an embedded clause in a sentence, we can extract the embedding predicate from the sentence by searching for the nearest VP parent of the clause. We iteratively search through the parents of the embedded clause until a VP parent is reached. We then identify the predicate span from this constituent, considering a wider range of possible verbs, adjectives, and prepositions than previous methods (cf. Section 2.1). For each constituent child of the VP (with exception to the final one which contains the embedded clause) we keep every token in the child span as long as the child label is either a PP, NP or SBAR label. For the last child of the VP, we keep every token until the onset of the embedded clause. We then filter these tokens based on their part-of-speech tags. We keep only the tokens that are VERB, ADP or ADJ, with the exception for an auxiliary tag AUX if there is also an adjective in the original token list. This helps us capture adjectival predicates such as 'unclear as to' or 'is certain' (see Figure 2).

Clause Identification Given that a sentence is detected as having an embedded clause, we can then further use the parsed representation to extract the span of the embedded clause. The constituency parser is advantageous in this regard as we take whatever is under the syntactic label of SBAR to be the embedded clause constituent.

Typing Having identified the clause span, the heuristics for typing the clause can involve more simple string matching. For alternative interrogative clauses we check the complementiser. If the complementiser *whether* is in the first word of the embedded clause along with the token *or*, then it is an alternative interrogative. If instead, we find *whether* that is not followed by the token *or* or is followed by the explicit string *or not*, then the embedded clause is a polar interrogative. If a unique token of either *which, who, what, when, where, why,* or *how* is the first word of the embedded clause, then it is a constituent interrogative clause. If none of the prior conditions are met, including if the clause begins with *that*, then we type the clause as declarative.

4.2 Evaluation

We evaluate the performance of our tool on the sentence annotations in GECS. With these annotations we can accurately test the tool's ability to detect embedded clauses, embedding predicates, and clause types, allowing us to evaluate how our tool handles messy natural data. We have also built a pattern matching baseline to compare our heuristics against a more linear approach.

Pattern Matching Baseline The baseline we constructed is a rule-based tool using pre-defined lexical patterns to extract embedded clause annotations from a sentence. This method relies on the SpaCy Matcher, a tool which is similar to Regular Expressions in that it matches a given pattern in a string, but with useful supplemental linguistic information encoded, such as POS and lemma (Honnibal and Montani, 2017)[7]. In order to de-

[7] We opted for a Regular Expression matcher baseline due to its straightforward implementation for this task, as com-

tect embedded clauses, the Matcher is provided with the fixed list of (potentially) embedding predicates from MegaAcceptability (White and Rawlins, 2016). It then returns instances of these predicates in a sentence; with an added heuristic ensuring that the predicate is followed by some other verb or auxiliary (i.e. a clause), an embedded clause is identified. Prepositions proceeding the verb are included in the list of predicate tokens and POS and lemmas are also identified. Limited by the linear nature of the Matcher, we define the clause span as the end of the predicate to the end of the sentence, ignoring any adverb/pronoun which may occur between a predicate and clause. For the final goal of typing the embedded clause we again use the Matcher to match the first token of the clause to the associated complementisers for each type. We distinguish between polar and alternative interrogatives by classifying clauses containing the token *or* but not the string *or not* as alternative, and every other instance as polar. If no associated complementiser is found in the clause, the clause is typed as declarative.

Detection	**Baseline**			**Parser Tool**		
	Precision	Recall	F1	Precision	Recall	F1
Single	0.54	0.94	**0.69**	0.90	0.94	**0.92**
Multi	0.74	0.85	**0.79**	0.94	0.83	**0.88**
Overall	0.54	0.91	**0.68**	0.90	0.91	**0.91**

Table 1: Precision, Recall, and F1 scores for clause detection in GECS.

Task	**Baseline**	**Parser Tool**
Predicate Identification	0.79	0.91
Clause Identification	0.50	0.87
Type Identification	0.94	0.96

Table 2: Identification accuracy scores evaluated on the true positives set of detected clauses from Table 1.

Results We split the evaluation of our tool and the baseline into three detection sections: *Single Clause Evaluation* which evaluates detection performance on sentences in GECS that only included one embedded clause, *Multi Clause Evaluation* which evaluates detection performance on sentences in GECS that had multiple embedded clauses (nested and coordinated clauses), and *Overall* which combines the statistics of single and multi clause evaluation and performance on the adversarials. Table 1 provides the precision and recall for these metrics. We also evaluated amongst the correctly detected embedded clauses the annotation abilities of our tool and the baseline, by seeing if the selected predicate is correct (*Predicate Identification*), if the selected clause is correct (*Clause Identification*), and if the typing of the clause is correct (*Type Identification*). Table 2 provides the accuracy scores for these metrics.

As Table 1 and 2 shows, we outperform the baseline in every metric, indicating that our method of utilising a constituency based tool is better than a linear based approach. Our tool only slightly degrades in detection recall when given a sentence that had nested and/or coordinated embedded clauses.

Failure analysis In the few cases of our tool's error, we see the following categories: parser errors, unconditionals mistaken as embedded clauses, and incomplete complex predicate detection. The parser error was the biggest issue for failed cases - unfortunately this is an unavoidable error given that any parser will be imperfect. Unconditionals also proved a problem because they are parsed the same as an embedded clause, and therefore are impossible to differentiate from one another. Finally, complex predicates were sometimes incompletely detected so not all of the predicate tokens are placed in the entry. Given that some of these errors are unavoidable, coupled with the tool's high precision and recall, we still take the results to indicate that our tool can be used to create a large-scale dataset of naturally-occurring embedded clauses, as long as researchers propagate the error into their analysis - something which needs to be done with any corpus study.

5 Case Study: Large-Scale Dataset

Having designed a tool which can identify and annotate embedded clauses, we applied it to an English corpus to create a large-scale dataset of annotated embedded clauses. We chose to apply the tool to a subset of *Dolma*[8] (Soldaini et al., 2023). Overall, $28,968,073$ embedded clauses were detected.

5.1 Comparison with MegaAcceptability

In order to compare with MegaAcceptability, we performed a limited case study on our large-scale dataset by only looking at our dataset entries that

pared with a Dependency Parser for instance.

[8] We extracted text from the the Dolma subset *v1_6-sample*.

included the 1007 verbs that were used in the MegaAcceptability templates. To get the rating of each verb from MegaAcceptability, we selected the maximum normalised rating of that verb's available constructions. We compared the distribution between the acceptability rating of a verb according to MegaAcceptability and its frequency in the large-scale dataset. It would be generally expected that the higher a verb is rated the more frequent it would be. As shown in Figure 3, this is the overall trend that we see. This means that our tool has successfully captured the verbs with the highest acceptability, while the verbs with lower acceptability had a lesser chance of occurring with embedded clauses.

There are some exceptions to the frequency-acceptability distribution, however this provides an interesting exploration point. For instance, the low acceptability outlier which has a high frequency in Figure 3 is the predicate *mean*. Looking at entries with *mean* as the predicate, we see three example types: (i) where it is unclear if the predicate is actually embedding or is acting as some filler (4a), (ii) false positives (4b), and (iii) true embedded clauses (4c). Thus, *mean* could be an outlier because of false positives, or it could be an outlier due to a data-driven approach collecting sentence clause types which a template approach could not.

(4) a. It's pretty catchy, I mean who doesn't go ANN ANN and A SORE.

b. In Glosa it means "what I've just said".

c. This means [...], the ADA applies to you.

5.2 Clauses and Predicates at Scale

With our large scale dataset of embedded clauses we can look beyond the fixed list of predicates as would be provided by a template-driven dataset like MegaAcceptability. With our approach we are able to view the clause-predicate distribution at a grand scale to test and verify linguistic theories. From the nearly 29 million embedded clause examples in the dataset we have the following distribution of clause types: $19,195,112$ declarative clauses, $9,402,868$ constituent interrogative clauses, $261,274$ polar interrogative clauses, and $108,819$ alternative interrogative clauses. This shows us how rare polar and alternative interrogative clauses are. Moreover, we can examine to the distribution of embedding predicates in the dataset. Taking a look at the part-of-speech tags for each of the embedding predicates in the dataset we can observe the distribution of adjectival and verbal predicates. Adjectival predicates require an accompanying verb or auxiliary (e.g., *be happy*), so we look at complex predicates involving two or more tokens. We find that there are $35,294$ unique adjectives within these complex predicates. Meanwhile, for simple one-word verbal predicates, we find $29,654$ unique predicates. Altogether, this leaves us with a strong set of examples to analyse any clause-predicate distribution of interest.

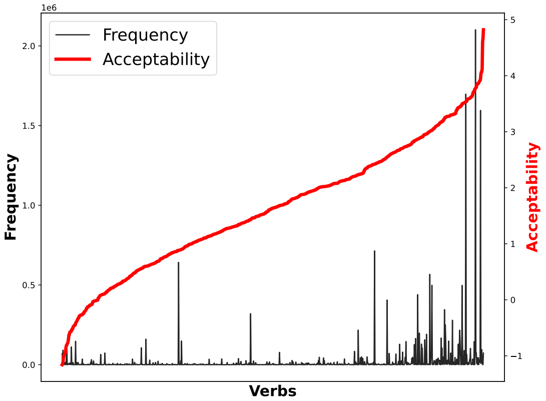


Figure 3: Comparison of natural data frequency and acceptability of the verbs found in MegaAcceptability ranked in increasing order of acceptability

Here we present an example of how the dataset can be used in linguistic research to further validate and verify linguistic theories, as well as survey new possible sentence constructions that could be of interest.

Emotive Factive Predicates As mentioned in Section 1, previous analyses have shown that emotive factive predicates, such as *be happy*, or *be glad*, are not able to embed either polar or alternative interrogative clauses (Karttunen, 1977; Abels, 2004; Sæbø, 2007). We can see if the extracted dataset shows this distribution statistically and if there are any counter-examples.

To test this generalisation, we selected a subset of emotive factive predicates to investigate further: *happy, amazed, sad, glad, excited, surprised, incredible, angry, mad, jealous, afraid*. Looking at the clauses that are embedded with these predicates, we get the following distribution of clause types: $175,479$ declarative clauses, $47,877$ constituent interrogative clauses, 159 polar, and 134 alternative interrogative clauses. The statistical breakdown does match the generalisation, with declarative and constituent interrogative embedded clauses being the more popular embedded clause type. However, more importantly, there are some polar and alternative interrogative, of which we can search through to find potential counter-examples to the generalisation.

In searching through the polar and alternative interrogative embedded clauses, many are false positives, with the following four errors being indicative of the set: unconditionals (5a), wrong predicate span where the emotive factive is not the embedding predicate (5b), real embedded clauses but the sentence does not have the intended meaning required by the generalisation, e.g., *be afraid* is non-factive in (5c), and clausal adjuncts (5d).

(5) a. It's not your problem, because you're happy whether you're with him or doing stuff on your own.

b. I'm not sure how excited to get about this fund and whether he's just piggybacking on the Buffett name.

c. We are afraid whether it will be in Sindhi interest.

d. Meanwhile, people across the state are hair-on-fire mad over whether urban water users should be allowed to buy rural property simply for the water rights, and whether some water users should be allowed to sell their water to others out of state.

Given that we need to propagate the tool's error rate, this is to be expected. However, there appears to be some genuine counter-examples (6), of which at least two of the three native speakers among the authors find grammatical. It is beyond the scope of this paper to provide an analysis of these sentences, so we leave it for future work.

(6) a. In the post you talk about your child's health issues and in the end ask if people are happy with whether they're circumcised or not.

b. You might be surprised about whether there's hope for future shooters.

Although this analysis is not exhaustive in the least, we use these examples to motivate the use of this dataset to further validate and explore linguistic theories through naturally-occurring linguistic data in addition to handcrafted templatic examples.

6 Discussion

As this is the first method at extracting embedded clauses from natural language corpora, we set out some future research avenues to be undertaken. Firstly, clausal embedding extraction should be extended to other languages so that linguistic theories using such large corpora can have crosslinguistic validity. Given that the universal definition of a clausal complement is a complement to VP, we argue that a similar method to what we have described in the paper can be taken with other languages. The main changes would be to the fine-grain heuristics that we used for typing. Of course, our approach is subject to the limitations that follow from any corpus-based research, which introduces its own set of biases pre-existing in the corpora. It is also not always possible to scale this approach crosslinguistically, as the method relies on a given language having large enough corpora (which many do not). Nonetheless, this should not deter people from using the method with an applicable language, in complement with other approaches. Secondly, we recognise that there are other potential methods for extracting English clausal embeddings. One such technique is the use of an LLM, a method which we decided against given that an LLM is a blackbox, meaning a thorough error analysis would not be able to be conducted. To aid future development within this area, we have provided GECS to be used as a benchmark for this task.

7 Conclusion

The availability of large natural-language corpora has led to an opportunity for linguists to conduct large-scale language studies. However, extracting specific language constructions from such large corpora is a difficult endeavour. A particular construction in need of a large-scale corpus study is that of embedded clauses. Thus, we have made three contributions in aid of fulfilling this need. Firstly, we have created GECS, a small-scale dataset with fine-grained gold standard annotation of embedded clauses to be used as a benchmark for embedded clause extraction in English. Secondly, we created a tool which can be applied to English natural language corpora to detect and annotate embedded clauses. And finally, we provided a large-scale extracted set of English embedded clauses from the natural language corpus *Dolma* for the linguistic community to use.

Acknowledgements

This work was supported in part by the UKRI Centre for Doctoral Training in Natural Language Processing, and funded by the UKRI (reference: EP/S022481/1) to the first three authors and the UKRI Future Leaders Fellowship (reference: MR/V023438/1) to the last author. We would like to thank the three anonymous reviewers, Mark Steedman, and Aaron White for their insightful comments.

References

Klaus Abels. 2004. Why *surprise*-predicates do not embed polar interrogatives. *Linguistische Arbeitsberichte*, 81:203–222.

Ivano Caponigro. 2003. *Free not to ask: On the semantics of free relatives and wh-words cross-linguistically*. Ph.D. thesis, University of California, Los Angeles.

Wayne Cowart. 1997. *Experimental Syntax: Applying Objective Methods to Sentence Judgments*, 1 edition. SAGE Publications.

Jon Robert Gajewski. 2007. Neg-raising and polarity. *Linguistics and Philosophy*, 30:289–328.

Jane Grimshaw. 1979. Complement selection and the lexicon. *Linguistic inquiry*, 10(2):279–326.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Larry Horn. 1978. Remarks on neg-raising. *Syntax and Semantics*, 9.

Lauri Karttunen. 1977. Syntax and semantics of questions. *Linguistics and philosophy*, 1:3–44.

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.

Clemens Mayr. 2019. Triviality and interrogative embedding: context sensitivity, factivity, and neg-raising. *Natural language semantics*, 27(3):227–278.

Petr Pajas, Jan Štěpánek, and Michal Sedlák. 2009. PML tree query. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Kyle Rawlins. 2008. *(Un)conditionals: An investigation in the syntax and semantics of conditional structures*. Ph.D. thesis, University of California, Santa Cruz.

Phillip Resnik and Aaron Elkiss. 2005. The linguist's search engine: An overview.

Kjell Johan Sæbø. 2007. A whether forecast. In *Logic, Language, and Computation: 6th International Tbilisi Symposium on Logic, Language, and Computation, TbiLLC 2005 Batumi, Georgia, September 12-16, 2005. Revised Selected Papers 6*, pages 189–199.

Carson T. Schütze. 2016. *The empirical base of linguistics : grammaticality judgments and linguistic methodology*. Language Science Press, Berlin.

Micah Shlain, Hillel Taub-Tabib, Shoval Sadde, and Yoav Goldberg. 2020. Syntactic search by example. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 17–23.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Pete Walsh, Hannaneh Hajishirzi, Noah A. Smith, Luke Zettlemoyer, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2023. Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv preprint*.

Nadine Theiler, Floris Roelofsen, and Maria Aloni. 2019. Picky predicates: why believe doesn't like interrogative complements, and other puzzles. *Natural language semantics*, 27(2):95–134.

Wataru Uegaki and Yasutada Sudo. 2019. The* hope-wh puzzle. *Natural Language Semantics*, 27(4):323–356.

Henk van Riemsdijk. 2006. Free relatives. In Martin Everaert and Henk van Riemsdijk, editors, *The Blackwell companion to syntax*, volume 2, pages 338–382. Blackwell, Oxford.

Aaron Steven White. 2021. On believing and hoping whether. *Semantics and pragmatics*, 14(6):1–21.

Aaron Steven White and Kyle Rawlins. 2016. A computational model of s-selection. *Proceedings from Semantics and Linguistic Theory*, 26(26):641–663.

Aaron Steven White and Kyle Rawlins. 2020. Frequency, acceptability, and selection: A case study of clause-embedding. *Glossa (London)*, 5(1):1–40.

Hedde Zeijlstra. 2018. Does neg-raising involve neg-raising? *Topoi*, 37(3):417–433.

Richard Zuber. 1982. Semantic restrictions on certain complementizers. In *Proceedings of the 13th International Congress of Linguists*, pages 434–436. Tokyo.

A Data Structure

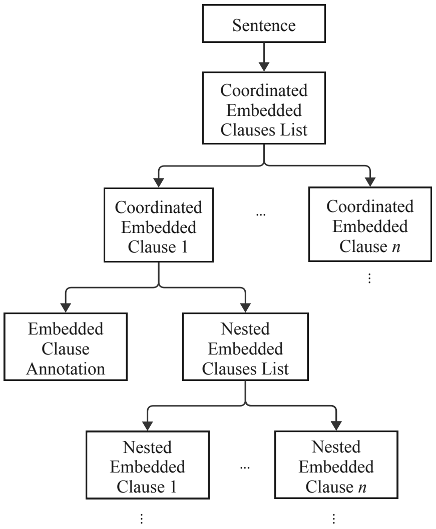

Sentence
Coordinated Embedded Clauses List
Coordinated Embedded Clause 1
...
Coordinated Embedded Clause *n*
Embedded Clause Annotation
Nested Embedded Clauses List
Nested Embedded Clause 1
...
Nested Embedded Clause *n*

Figure 4: The data structure in GECS for coordination and nesting of embedded clauses.

B Examples of Parser Errors

We have provided some examples of parser errors which are discussed in section 4.2. The sentence in Figure 5 is usually interpreted as a conditional sentence, however the parser represented it as an embedded alternative interrogative clause. On the other hand, the sentence in Figure 6 is an embedded clause, but the parser represented it as a relative clause.

C Neg-Raising Generalisation

Another linguistic generalisation that concerns embedded clauses is Neg-Raising. When syntactically negated, Neg-Raising predicates can take semantically higher scope than negation (Horn, 1978; Gajewski, 2007; Zeijlstra, 2018). Consider (7a), which can have the reading as in (7a) but can also have the interpretation in (7b). Accordingly, *believe* is described as a Neg-Raising predicate.

(7) a. I don’t believe John is nice

b. I believe that John isn’t nice

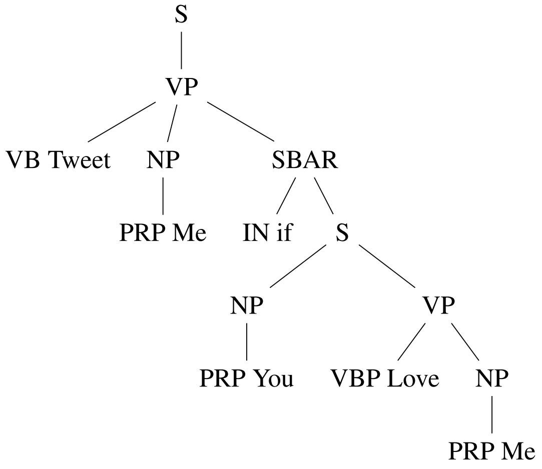


Figure 5: False-positive parse of a sentence with a conditional

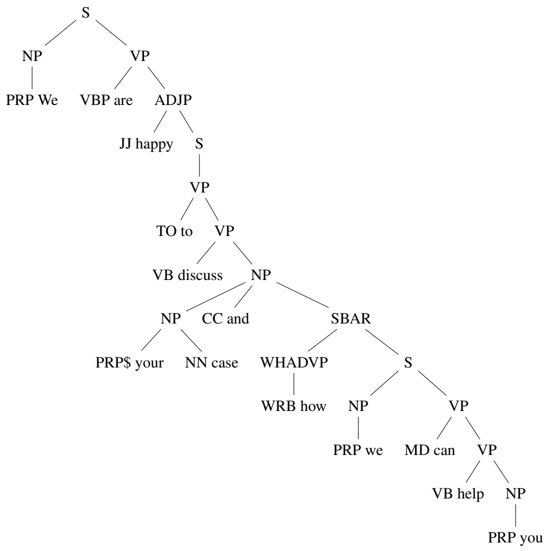


Figure 6: False-negative parse of a constituent embedded clause

However, many other embedding predicates do not allow this reading. Consider *predict* in (8a), which does not have the reading in (8b), so that *predict* is a non-Neg-Raising predicate

(8) a. I don't predict it will rain

b. I predict it will not rain

A generalisation of Neg-Raising predicates that has come from the literature is that they do not select for interrogative embedded clauses (Zuber, 1982; Mayr, 2019; Theiler et al., 2019). Thus, we can see in our dataset if there are genuine counter-examples to this generalisation, by searching for the Neg-Raising predicates that have interrogative embedded clauses. We did this with the predicate *believe* which occurred $497,684$ amount of times in the dataset, of which $19,787$ were interrogative embedded clauses. Looking at a subset of these interrogative embedded clauses, the majority of the instances were not true embedded clauses, which is to be expected as we need to propagate the error of the tool into our analysis. However, there were a few potential true counter-examples like (9), which two of the three native speakers amongst the authors found grammatical. However, we leave their analysis for future work.

(9) a. As we are not omniscient, we can't validate/believe (absolutely) whether God's existence is true/absolute.

b. Ernst was also asked if she believed whether or not the Russia investigation was warranted.

Automata for subregular syntax: Syntax with strings attached

Thomas Graf and **Kenneth Hanson**
Department of Linguistics
Stony Brook University
mail@thomasgraf.net and mail@kennethhanson.net

Abstract

Building on recent work in subregular syntax, we argue that syntactic constraints are best understood as operating not over trees, but rather strings that track structural relations such as dominance and c-command. Even constraints that seem intrinsically tied to trees (e.g. constraints on tree tiers) can be reduced to such strings. We define *serial constraints* as an abstraction that decomposes string constraints into a *context function* (which associates nodes with strings) and a *requirement function* (which enforces constraints on these strings). We provide a general procedure for implementing serial constraints as deterministic tree automata. The construction reveals that the many types of constraints found in subregular syntax are variants of the same computational template. Our findings open up a string-based perspective on syntactic constraints and provide a new, very general approach to the automata-theoretic study of subregular complexity.

1 Introduction

One of the most common assumptions in theoretical and computational linguistics alike is that syntax does not operate over strings but rather trees, DAGs, or even more complex structures. This is the case for all major syntactic formalisms, including a.o. Minimalism, HPSG, LFG, TAG, and CCG. Even in formal language theory, where many findings focus on the complexity of syntax as a set of well-formed strings (Huybregts, 1984; Kornai, 1985; Shieber, 1985; Radzinski, 1991; Michaelis and Kracht, 1997; Kobele, 2006, a.o.), there is a large body of work that analyzes these strings as the yield of tree structures (e.g. the characterization of multiple context-free string languages as the string yields of MSO-definable tree languages under an MSO tree-to-tree transduction). But even though syntax may well do a lot of work with richly structured objects, this does not entail that this structure is readily accessible to all parts of syntax. To the contrary, recent work in *subregular syntax* (Graf, 2022a,b) suggests that syntactic constraints are so limited that they are better understood as operating over strings, albeit strings that encode linguistic relations like dominance and c-command rather than linear precedence (cf. Frank and Vijay-Shanker, 2001).

For example, Principle A of binding theory requires a reflexive such as *herself* to be c-commanded by a compatible DP like *Mary* or *the woman* within a specific locality domain. As explained in Graf and Shafiei (2019), enforcing Principle A does not require access to the full tree structure as we only need to know the list of c-commanders of the reflexive, which can be represented as a string. Even wh-movement, one of the most fundamental aspects of syntax, can be understood as a constraint that a wh-landing site imposes on its string of *wh-tier daughters* (Graf, 2022a, p.275f). Thus, while syntax may build tree structures for use at the interfaces (meaning, prosody), its constraints appear to be limited to particular types of strings that do not provide nearly as much information as the tree they are obtained from.

This paper puts this general observation on a formal foundation. We introduce the notion of *serial constraints*, which are pairs consisting of a *context function* and a *requirement function* (Sec. 2). The context function con associates every node n of tree t with a string that encodes its syntactic context in t, e.g. its string of ancestors or its string of wh-tier daughters. The requirement function req maps each n to a string language. Then t is well-formed with respect to the serial constraint iff it holds for every node n of t that $con(n) \in req(n)$. We argue that all the proposals put forward in the subregular syntax literature so far are instances of serial constraints (Sec. 2.3–2.5). We then show how serial constraints can be implemented as deterministic tree automata (Sec. 3). For some constraints,

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 333-345.
Eugene, Oregon, July 18-20, 2025

this takes the form of deterministic bottom-up tree automata (Sec. 3.1, 3.2), while for others it takes the form of sensing tree automata, which are deterministic top-down automata with a look-ahead of 1 (Sec. 3.3, 3.4). Despite that difference in directionality, the automata follow a common construction that can be expressed in algebraic terms as a formula of Boolean matrix multiplication steps. These formulas can be tweaked in various ways to define new types of string representations, opening up a novel perspective on subregular automata for syntax.

Our findings have several implications. First of all, our framework provides the first automata-theoretic description of tier-based strictly local tree languages. While there has been a lot of work on tier-based strict locality for strings (Lambert and Rogers, 2020), extending it to trees is not trivial. Since a node can have arbitrarily many tier daughters, one cannot simply store them all in the states of the tree automaton. Our automata construction resolves this challenge and might even provide a new foundation to develop a subregular theory of tree automata. Second, serial constraints formally link two branches of subregular syntax that seem to have been moving in different directions: tree tiers with local constraints VS strings with tier-based strictly local constraints. Our findings reassert the status of subregular syntax as a unified program that furnishes computationally restricted yet linguistically flexible ways of analyzing syntactic phenomena. Finally, the reduction of syntactic constraints from trees to strings opens up new attack vectors for syntactic learning. For example, neural networks could be trained on corpora that lack full tree structures but include relevant c-command relations, encoded as a string.

It is also important to emphasize what this paper is not about. We do not claim that tree structure is redundant for syntax. As mentioned above, the structure-building aspect of syntax seems crucial for prosody and semantic interpretation. Following the two-step approach (Morawietz, 2003; Mönnich, 2006), we regard syntax as the interaction of two components: syntactic constraints that define the set of well-formed structures, and a transductive component that maps syntactic objects to output structures that are used at the PF and LF interfaces. We are not currently aware of any method to reduce the latter to strings, and even subregular work on the transductive component presupposes trees for this (Graf, 2023). But syntactic constraints are amenable to such a reduction and all the methodological simplifications this may provide — as long as strings are built over pertinent syntactic relations rather than linear precedence.

2 Serial constraints for syntax

We take as our starting point recent proposals from subregular syntax (see Graf 2022a,b for a recent overview). In subregular syntax, syntactic structures are feature-annotated dependency trees that encode derivations of a variant of Minimalist grammars (MGs; Stabler, 1997, 2011) where licensee features are unordered (Sec. 2.1; see also appendix A for additional background on MGs and their dependency trees). These syntactic structures are regulated by various subregular constraints, and we define *serial constraints* (Sec. 2.2) as a general mechanism that unifies the many proposals in the subregular literature (Sec. 2.3–2.5). Serial constraints could also be used with other kinds of tree structures, but this paper limits itself to the kind of MG dependency trees used in the subregular literature.

2.1 MG derivations as dependency trees

We treat trees as *labeled Gorn domains* (Gorn, 1967), but for convenience we assume that daughters are numbered from right to left. A *Gorn address* is a string of natural numbers ($s \in \mathbb{N}^*$), including the empty string ε. A *Gorn domain* D is a set of Gorn addresses such that I) $ui \in D$ implies $u \in D$ for all $u \in \mathbb{N}^*$ and $i \in \mathbb{N}$ (mother-of closure), and II) $uj \in D$ implies $ui \in D$ for all $u \in \mathbb{N}^*$, $i, j \in \mathbb{N}$, and $i < j$ (right sibling closure). We occasionally use ux to refer to the unique address $ui \in D$ such that $u \in \mathbb{N}^*$, $i \in \mathbb{N}$, and $u(i+1) \notin D$. A Σ*-tree* is a pair $\langle D, \ell \rangle$ where D is a Gorn domain and $\ell : D \to \Sigma$ the (total) *labeling function*. When clear from context (and particularly in Sec. 3), we use the term *node* to refer to either a Gorn address or its label.

Let Lex be an MG lexicon, i.e. a finite set of lexical items and thus an alphabet. We call a Lex-tree an **MG dependency tree** (MDT) over Lex. Given node u of MDT t, node ui is interpreted as the i-th argument of u (see Fig. 1). Since there is a fixed upper bound j on the number of arguments a lexical item may take, we may assume w.l.o.g. that there is a fixed bound $k \geq j$ such that every Lex-tree is at most k-ary branching. Limited branching is crucial for our automata implementation in Sec. 3.

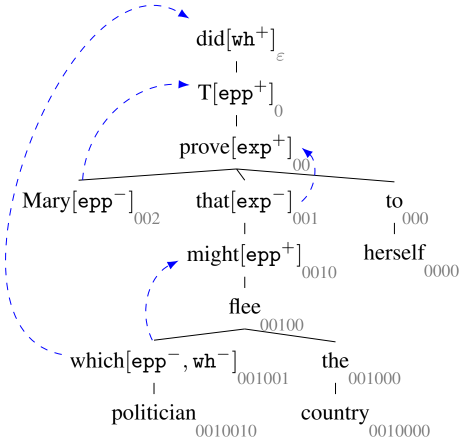


Figure 1: Dependency tree for **Which politician did Mary prove to herself that might flee the country?*, with Gorn address subscripts and dashed movement arrows added as visual aids

Even though MGs make heavy use of movement, all phrases in an MDT remain in their base positions. Movement is indicated via movement features, with the actual displacement left to a post-syntactic transduction step. Negative features (e.g. wh$^-$, epp$^-$) mark the head of the moving phrase, and positive features (e.g. wh$^+$, epp$^+$) mark the head that provides the corresponding landing site. For additional background on the linguistic interpretation of MDTs, the reader is referred to appendix A.

2.2 Serial constraint = context + requirement

We define string-based constraints on trees as the interaction of two functions. The **context function** defines a system for mapping each node n of a tree t to a specific string that is derived from the structural relations of t, relative to n. In some cases, a set Ω of diacritic symbols is used to distinguish multiple structural relations within the string. The **requirement function** then regulates the shape of the string n is mapped to. While all kinds of requirement functions could be considered, the proposals from the subregular literature can be captured with maximally simple ones that constrain the string of n based solely on the label of n. Combining a context function with a requirement function yields a **serial constraint**.

Definition 1. An **Ω-augmented context function** over Σ is a total function *con* that takes as inputs a Σ-tree $\langle D, \ell \rangle$ and Gorn address $a \in D$ and maps them to a (possibly empty) string $\langle \ell(a_1), \omega_1 \rangle \cdots \langle \ell(a_n), \omega_n \rangle$ such that $n \geq 0$ and for all $0 \leq i \leq n$, both $a_i \in D$ and $\omega_i \in \Omega$. If $|\Omega| = 1$, the context function is **unaugmented**. ⌟

To avoid visual clutter, we write $\ell(a)^\omega$ instead of $\langle \ell(a), \omega \rangle$, and we completely omit any mention of Ω for unaugmented context functions. To further increase readability, we use $\cdot$ to explicitly separate the symbols in the outputs of context functions.

Example 1. The (unaugmented) *daughter string* context function *drs* maps every node to its string of daughters, ordered from left to right. In Fig. 1, *drs*(prove[exp$^+$]) is Mary[epp$^-$]$\cdot$that[exp$^-$]$\cdot$to. ⌟

Definition 2. An **Ω-augmented requirement function** over alphabet Σ is a total function *req* : $\Sigma \to \wp((\Sigma \times \Omega)^*)$ that associates every symbol with a (possibly empty) string language over $\Sigma \times \Omega$. We say that *req* is **regular** iff *req*(σ) is a regular string language for every $\sigma \in \Sigma$. ⌟

Again we will use superscripts instead of pair notation, and we will omit Ω for unaugmented requirement functions. Hence the co-domain of unaugmented *req* is simplified to $\wp(\Sigma^*)$.

Example 2. The requirement function *Merge* maps every lexical item to its set of possible argument configurations, each one represented as a string. For example, the transitive verb *eat* is mapped to the set L_{DD} of all strings consisting of exactly two lexical items that each are of category D. Intransitive *eat* would instead require exactly one such D (and thus its image under *Merge* is L_D). If the grammar formalism does not disambiguate between the two, then *eat* is mapped to $L_{DD} \cup L_D$. ⌟

Definition 3. An Ω-augmented **serial constraint** over Σ is a pair $\langle con, req \rangle$ such that *con* is an Ω-augmented context function over Σ and *req* is an Ω-augmented requirement function over Σ. A Σ-tree $t := \langle D, \ell \rangle$ is well-formed with respect to $\langle con, req \rangle$ iff it holds for every $a \in D$ that $con(t, a) \in req(\ell(a))$. ⌟

Example 3. Selectional restrictions of lexical items can be regarded as a serial constraint that combines the context function *drs* over MDTs with the requirement function *Merge*. ⌟

2.3 Types of context functions: a-strings

We now turn to how the various string representations used in the subregular literature (Graf and Shafiei, 2019; Shafiei and Graf, 2020; Graf, 2022a) can be reconceptualized as context functions. We start our discussion with a-strings as they are the most intuitive.

Definition 4 (a-string). Given Gorn address u of MDT $t := \langle D, \ell \rangle$, the **a[ncestor]-string** context function as maps t and u the string of nodes in t that properly dominate u (in top-down order):[1]

$$as(t,u) := \begin{cases} \varepsilon & \text{if } u = \varepsilon \\ as(t,v) \cdot \ell(v) & \text{if } u = vi, i \in \mathbb{N} \end{cases}$$

Example 4. The a-string of which$[\texttt{epp}^-, \texttt{wh}^-]$ in Fig. 1 is did$[\texttt{wh}^+]$·T$[\texttt{epp}^+]$·prove$[\texttt{exp}^+]$ ·that$[\texttt{exp}^-]$ ·might$[\texttt{epp}^+]$·flee. ⌟

A-strings can be used to enforce constraints on movement paths. This includes domain conditions like island constraints, but also morphological alternations triggered by movement, e.g. wh-agreement in Irish (McCloskey, 2001; Georgi, 2017; Graf, 2022c).

Example 5. If a subordinate clause is headed by *that*, then its subject cannot be extracted out of this clause. This is known as the *that*-trace effect. This constraint is violated by *which politician* in Fig. 1. We can model this with the context function as in combination with a fairly simple requirement function req. If n carries the subject movement feature $\texttt{epp}^-$ then one of the following must hold: the rightmost complementizer in $as(n)$ is not *that*, or for every movement feature $\texttt{f}^-$ of n, at least one $\texttt{f}^+$ occurs in $as(n)$ to the right of the rightmost complementizer. If n does not include $\texttt{epp}^-$, $req(n)$ is Σ^*. The MDT in Fig. 1 is ill-formed because $as(which[\texttt{epp}^-, \texttt{wh}^-])$ is rejected by req due to the rightmost complementizer being *that* with no $\texttt{wh}^+$ occurring after it. ⌟

2.4 Types of context functions: c-strings

Whereas a-strings are mostly used to capture effects related to movement, c-strings track licensing requirements that are mediated by c-command.

Definition 5 (c-string). Given Gorn address u of MDT $t := \langle D, \ell \rangle$, its **c[ommand]-string** context function cs is recursively defined as[2]

$$cs(t,u) := \begin{cases} \varepsilon & \text{if } u = \varepsilon \\ cs(t,v) \cdot \ell(v) & \text{if } u = vx \\ cs(t,vi) \cdot \ell(vi)^{\leftarrow} & \text{if } u = v(i-1) \end{cases}$$

Example 6. The c-string of which$[\texttt{epp}^-, \texttt{wh}^-]$ in Fig. 1 is did$[\texttt{wh}^+]$·T$[\texttt{epp}^+]$·prove$[\texttt{exp}^+]$ ·Mary$[\texttt{epp}^-]^{\leftarrow}$·that$[\texttt{exp}^-]$·might$[\texttt{epp}^+]$·flee.
The c-string of *herself* is did$[\texttt{wh}^+]$·T$[\texttt{epp}^+]$ ·prove$[\texttt{exp}^+]$·Mary$[\texttt{epp}^-]^{\leftarrow}$·that$[\texttt{exp}^-]^{\leftarrow}$·to. ⌟

Intuitively, the c-string of n is obtained by traversing the tree from n towards the root in a leftmost manner, never moving right or down. This approximates the linguistic notion of c-command but does not track how movement may create new c-command relations or destroy existing ones (but we believe that the automata-theoretic view in Sec. 3 furnishes the right tools for addressing this in the future). In addition, c-strings also make an explicit distinction between containing c-commanders (X) and non-containing c-commanders ($\text{X}^{\leftarrow}$), which is crucial for some constraints such as Principle A. C-strings are our only instance of such an augmented context function.

Example 7. Consider a simplified version of Principle A: if n is a reflexive, then the smallest TP containing n must contain a DP that c-commands n. In our framework, this means that $cs(t,n)$ must contain some $X^{\leftarrow}$ such that X carries category feature D and occurs to the right of the rightmost T in the string. If n is not a reflexive, the Principle A requirement function PrA puts no restrictions on it (we set $PrA(n) := \Sigma^*$). ⌟

Note that for every node n of any MDT t, $as(t,n)$ is the longest subsequence of $cs(t,n)$ that does not contain any symbols with the superscript $^{\leftarrow}$. In subregular terms, $as(t,n)$ is a *tier* of $cs(t,n)$. It follows that every regular constraint over a-strings can be restated as a regular constraint over c-strings. Hence our automata-theoretic treatment of c-strings in Sec. 3 is also an implicit treatment of a-strings.

2.5 Types of context functions: T-strings

Our last and perhaps most abstract type of strings is defined via tree tiers. Intuitively, a tree tier $T(t)$ of tree t is constructed by fixing a set of node labels,

[1] The definition in Shafiei and Graf (2020) uses a bottom-up order for a-strings, which is formally equivalent but less elegant for our purposes. Moreover, Shafiei and Graf always include $\ell(n)$ in $as(n)$ in order to track which constraints should apply to the string. Since our approach leaves constraint selection to req, including $\ell(n)$ in $as(n)$ is redundant. In fact, factoring out constraint selection reduces the complexity of the string constraints in Shafiei and Graf (2020) from IOTSL to OTSL.

Note that the same differences also hold for the definition of c-strings in Graf and Shafiei (2019) and our Def. 5.

[2] The original definition in Graf and Shafiei (2019) does not include the augmentation symbol $^{\leftarrow}$.

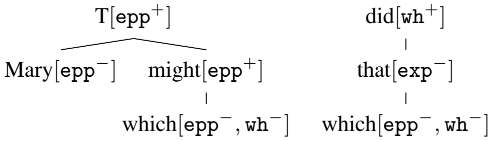


Figure 2: Two tiers of the MDT in Fig. 1: epp-tier (left) and *that*-trace tier (right)

the *tier alphabet* T, and removing from t all nodes that do not belong to T. Figure 2 shows two tiers of the MDT in Fig. 1. The epp-tier is obtained by removing all nodes whose label does not include $\texttt{epp}^+$ or $\texttt{epp}^-$, whereas the *that*-trace tier keeps all instances of the complementizer *that* and all nodes that carry $\texttt{wh}^+$ or $\texttt{wh}^-$. On a tree tier, the label of a node determines the shape that its string of daughters must have. Hence tree tiers are a visual metaphor for a context function that associates every node with its string of tier daughters.

Definition 6 (Tier strings). Given Gorn address u of MDT $\langle D, \ell \rangle$ and $T \subseteq \Sigma$, we say that u is *on* T iff $\ell(u) \in T$. Then u is the T*-mother* of v (and v is a T*-daughter* of u) iff u and v are both on T, $v = uv_1 \cdots v_n$ ($v_i \in \mathbb{N}$, $n \geq 1$), and for all $1 \leq i < n$ it holds that $uv_1 \cdots v_i$ is not on T. Furthermore, w T*-precedes* w' iff w and w' have the same T-mother and there exist $u, v, v' \in \mathbb{N}^*$ and $i, j \in \mathbb{N}$ such that $i > j$, $w = uiv$, and $w' = ujv'$.

The T**-string** context function T maps t and u to the set of T-daughters of u in t, ordered by T-precedence. When u has no T-daughters, $T(n) := \varepsilon$. ⌟

Example 8. Let epp be the alphabet of the epp-tier, which includes all labels that contain $\texttt{epp}^+$ or $\texttt{epp}^-$, and wh the corresponding alphabet for the wh-tier. Then the epp-string of which[$\texttt{epp}^-, \texttt{wh}^-$] in Fig. 1 is ε, and so is its wh-string. The epp-string of T[$\texttt{epp}^+$] is Mary[$\texttt{epp}^-$]·might[$\texttt{epp}^+$]. The wh-string of did[$\texttt{wh}^+$] is which[$\texttt{epp}^-, \texttt{wh}^-$], whereas its *that*-trace string is that[$\texttt{exp}^-$]. ⌟

Like a-strings, tier strings can be used to enforce island constraints and other conditions on individual movement paths. In contrast to a-strings, they also capture constraints on how distinct movement paths may interact.

Example 9. In MGs, every landing site must be targeted by exactly one mover. This can be captured over tier strings: for every n with movement feature $\texttt{f}^+$ it must be the case that the f-string of n contains exactly one lexical item with $\texttt{f}^-$. A-strings, by contrast, can enforce the presence of a landing site for a mover (if n carries $\texttt{f}^-$, then $as(n)$ must contain $\texttt{f}^+$) but cannot guarantee that this landing site isn't targeted by multiple movers (e.g. C[$\texttt{wh}^+$] when the subject and object both carry $\texttt{wh}^-$, as neither one appears in the other's a-string).

Daughter strings as defined in example 1 are identical to T-strings with $T = \Sigma$. Hence our automata-theoretic treatment of T-strings also subsumes daughter strings.

3 Tree automata for serial constraints

The previous section has identified several string representations and constraints that have been invoked in the subregular literature, and we have recast all of them as context functions that can be combined with suitable requirement functions. Since the constraints from the subregular literature all define subregular string languages, they can all be captured with regular requirement functions and thus finite-state string automata (FSAs). While requirement functions are easily understood and implemented, then, the formal status of context functions is less clear.

We propose to model context functions as deterministic tree automata whose only purpose is to decide which nodes should be fed as input to the requirement function. These tree automata simulate the (FSAs of the) requirement functions in their state space, while different context functions correspond to minimally different matrix multiplication formulas for updating states. This has the advantage that the tree automaton implicitly produces and evaluates in a single run all n string representations that the corresponding context function would produce for a tree with n nodes. The matrix multiplication formulas also provide a very general template that can be easily adapted to new kinds of string representations.

3.1 Automata for T-string context functions

T-string context functions are implemented as (deterministic) bottom-up tree automata. We present the general template here and provide an illustrative example in Sec. 3.2. Our construction assumes that regular requirement functions are decomposed into FSAs, one per symbol in the alphabet. The FSAs are subsequently decomposed into Boolean matrices. As mentioned in the introduction, this addresses a central challenge of dealing with T-strings: even when MDTLs are assumed to be at most k-ary branching, there is no upper bound on the number of T-daughters a node may have. Hence one cannot represent the entire string of T-daughters in the states of the tree automata. Instead, one has to store how the T-daughters seen so far would cause the FSA of $req(\sigma)$ to transition between states. The matrix representation of FSAs makes this very easy. Readers unfamiliar with this construction are referred to Appendix B for additional background.

A bottom-up tree automaton is a 4-tuple $\mathfrak{A} := \langle \Sigma, Q, F, \Delta \rangle$ where Σ is an alphabet, Q is a finite set of states, $F \subseteq Q$ is the set of final states, and Δ is a set of transitions. Transitions are of the form $\sigma(q_1, \ldots, q_n) \Rightarrow q$ ($\sigma \in \Sigma$, $q_i \in Q$, $n \geq 0$).[3] Intuitively, the automaton processes trees from the leaves to the root, assigning each node n a state $q \in Q$ based on I) the label of n, and II) the states of n's daughters. The automaton recognizes tree t iff the root of t is assigned some $q \in F$.

Let $\mathcal{B}$ be the Boolean matrix representation of some FSA with $m \geq 1$ states that generates the string language $req(\sigma)$, where $\sigma \in \Sigma$ and req is some regular, Ω-augmented requirement function. We use $\mathbf{I}$ for the initial matrix, $\mathbf{F}$ for the final matrix, $\mathbf{b}(\sigma)$ for the Boolean matrix corresponding to symbol $\sigma \in \Sigma \times \Omega$, and $\mathbf{id}_m$ for the identity element for matrix multiplication of Boolean $m \times m$ matrices. We use $\otimes$ to denote Boolean matrix multiplication.

For every $\sigma \in \Sigma$, we construct a bottom-up tree automaton $\mathfrak{A}_\sigma$ that ensures for every tree t and node n with $\ell(n) = \sigma$ that $T(t, n) \in req(\sigma)$. Intersecting all $\mathfrak{A}_\sigma$ for $\sigma \in \Sigma$ yields a bottom-up tree automaton that enforces requirement function req over T-strings.

Our construction automatically assembles $\mathfrak{A}_\sigma$ from the specification of just two attributes for each node label.

Definition 7 (Node attributes). Let $V, O \subseteq \Sigma$ be the set of **visible** and **opaque** nodes, respectively, and $\sigma \in \Sigma$ the **restricted** node. Then the *value* $\mathbf{v}(n^\omega)$ of $n^\omega \in \Sigma \times \Omega$ is $\mathbf{b}(n^\omega)$ if $n \in V$, and $\mathbf{id}_m$ otherwise. Given a Boolean matrix q, $q \oplus n^\omega$ is $\mathbf{v}(n^\omega)$ if $n \in O$ and $q \otimes \mathbf{v}(n^\omega)$ otherwise. Given initial matrix $\mathbf{I}$ and final matrix $\mathbf{F}$, $\mathbf{r}_n(q)$ is undefined if both $n = \sigma$ and $\mathbf{I} \otimes q \otimes \mathbf{F} = 0$; otherwise $\mathbf{r}_n(q) = q$. ⌟

Intuitively, visible nodes are those that can cause the underlying FSA to transition to a new state. For T-strings, those are simply the nodes that are on the tier. Opaque nodes induce locality domains by overwriting the result of previous matrix multiplications with their own value. For T-strings, every visible node is also opaque. The restricted nodes for $\mathfrak{A}_\sigma$ are exactly those labeled σ, i.e. the ones whose T-string must be well-formed according to the underlying FSA.

Definition 8 (T-string automaton). We define $\mathfrak{A}_\sigma := \langle \Sigma, Q_\mathcal{B}, Q_\mathcal{B}, \Delta \rangle$. Here $Q_\mathcal{B}$ is the result of closing the set of square matrices in $\mathcal{B}$ under Boolean matrix multiplication. For every $n \in \Sigma$ and all $q_1, \ldots, q_k \in Q_\mathcal{B}$ ($k \geq 1$), we set

$$q := \mathbf{r}_n \left(\bigotimes_{i=1}^{k} q_i \right) \oplus \mathbf{v}(n)$$

such that $\sigma(q_1, \ldots, q_k) \Rightarrow q \in \Delta$ iff q is defined. Furthermore $\sigma() \Rightarrow \mathbf{v}(\sigma) \in \Delta$ for every $\sigma \in \Sigma$. ⌟

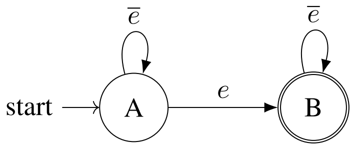
Since the formula above yields at most one value for q, $\mathfrak{A}_\sigma$ is deterministic even if $\mathcal{B}$ is the Boolean representation of a non-deterministic FSA. Note that $Q_\mathcal{B}$ and Δ are automatically constructed from σ, Σ, $\mathcal{B}$, and the attributes V and O. Also, all states of $\mathfrak{A}_\sigma$ are final (a tree is rejected iff there is a node that no state can be assigned to). Hence $\mathfrak{A}_\sigma$ could instead be defined as a 5-tuple $\langle \Sigma, \sigma, \mathcal{B}, V, O \rangle$.

3.2 Example: `epp`-string requirement function

Example 9 mentions that MGs require every landing site to be targeted by exactly one mover. If a node carries the feature $\texttt{epp}^+$, then its epp-string must contain exactly one node that carries $\texttt{epp}^-$. We can think of this as a requirement function $\mathbf{1}$ that maps each lexical item to one of two regular string languages. If l does not carry $\texttt{epp}^+$, then $\mathbf{1}(l)$ is Σ^*; otherwise, $\mathbf{1}(l)$ is $L_\mathbf{1} := \overline{E}^* E \overline{E}^*$ where $E \subseteq \Sigma$ is the set of lexical items that carry $\texttt{epp}^-$ and $\overline{E} := \Sigma - E$. It is easy to define an FSA for $L_\mathbf{1}$, which is then decomposed into its Boolean representation $\mathcal{B}$ (see Fig. 3).

We now construct $\mathfrak{A}$ for a single lexical item, which is $might[\texttt{epp}^+]$. The epp-tier contains all items that carry $\texttt{epp}^+$ or $\texttt{epp}^-$, so all of those are visible and opaque. Figure 4 shows the states assigned by the automaton as well as the attributes of

[3] In the tree automata literature, it is more common to write the transition rules in the format $\sigma(q_1(x_1), \ldots, q_n(x_n)) \Rightarrow q(\sigma(x_1, \ldots, x_n))$, with each x_i a variable representing a subtree (see Gécseg and Steinby 1997 and Comon et al. 2008, p.20). We omit these variables to reduce clutter.



$$\mathbf{I} := \begin{pmatrix} 1 & 0 \end{pmatrix} \qquad \mathbf{F} := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\mathbf{e} := \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad \overline{\mathbf{e}} := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \mathbf{id}_2 := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Figure 3: FSA for $L_\mathbf{1}$ and corresponding Boolean matrices, with $e/\overline{e}$ as a shorthand for every lexical item on the epp-tier that does/doesn't carry epp$^-$.

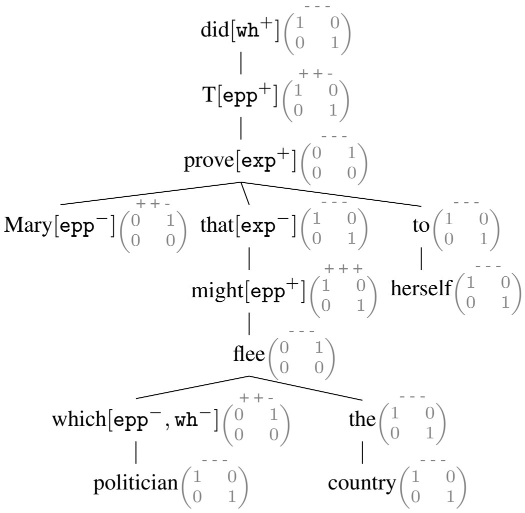

did[wh$^+$] - - - $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
T[epp$^+$] + + - $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
prove[exp$^+$] - - - $\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$
Mary[epp$^-$] + + - $\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$
that[exp$^-$] - - - $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
to - - - $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
might[epp$^+$] + + + $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
herself - - - $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
flee - - - $\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$
which[epp$^-$, wh$^-$] + + - $\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$
the - - - $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
politician - - - $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
country - - - $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

Figure 4: Run of automaton enforcing $L_\mathbf{1}$ over epp-strings of $might$[epp$^+$]; + and − indicate whether the node is visible, opaque, and/or restricted

each node (in the order *visible-opaque-restricted*). The tree is correctly recognized as well-formed with respect to the epp-string requirement function because the root is assigned a state (remember that all states are final).

Let us consider a few specific nodes from Fig. 4. The leaf node *politician* is not visible, so its value is the 2×2 identity matrix $\mathbf{id}_2$. Since it is a leaf, we use the transition $\sigma() \Rightarrow \mathbf{v}(\sigma)$, for which it is irrelevant whether *politician* is opaque or restricted. Now consider *which*, right above *politician*. It is visible and opaque, but not restricted. Since it is visible and carries epp$^-$, its value is the Boolean matrix $\mathbf{e}$. Since it isn't restricted, $\mathbf{r}_n(\bigotimes_{i=1}^{k} q_i)$ is not undefined, but since it is opaque its state is just its value $\mathbf{e}$. The node *flee* above *which* is neither visible nor opaque or restricted. Hence its state is the result of matrix multiplying the states of *which* and *the* with its value $\mathbf{id}_2$. Finally, *might* is visible, opaque, and restricted. Its value is the Boolean matrix $\overline{\mathbf{e}}$, which is identical to $\mathbf{id}_2$ in this case. As *might* is restricted, $r_n(\bigotimes_{i=1}^{k} q_i)$ could be undefined. But fortunately multiplying $\mathbf{I}$ with the state of *flee* and $\mathbf{F}$ yields 1 (confirming that the epp-tier string of *might* is a member of $L_\mathbf{1}$). If the result had been 0, no state would have been assigned and the computation would have halted, causing the tree to be rejected. Instead, *might* is assigned its value as its state because it is opaque. The computation continues from there, but since no other nodes in the tree are restricted, we are guaranteed to assign some state to the root (which is final because all states are final).

By constructing such an automaton for every lexical item with epp$^+$, we ensure that every lexical item with epp$^+$ has exactly one epp$^-$ among its epp-tier daughters.

3.3 Automata for c-string context functions

We now turn to the implementation of c-strings (which subsumes a-strings as discussed at the end of Sec. 2.4). Instead of bottom-up automata, we will use *sensing tree automata* as these have previously been proposed by Graf and De Santo (2019) as a model of c-string constraints.[4] For convenience, we use a slightly different notation for defining the transitions of these automata, and we allow the initial state to be determined by the label of the root. These changes will make it easier to see that the state assignment template for sensing tree automata is almost the same as for bottom-up tree automata. In particular, the attributes and operations from Def. 7 carry over unaltered. This shows that our treatment of context functions is independent of the specific types of tree automata.

A sensing tree automaton is a 3-tuple $\mathfrak{A} := \langle \Sigma, Q, \delta \rangle$ where Σ is an alphabet, Q is a finite set of states, and δ is a set of transition rules that may take two distinct forms. For interior nodes, we have $\langle q, \sigma(\sigma_1, \ldots, \sigma_k), i \rangle \Rightarrow q_i$ $(1 \leq i \leq k)$. This means that if σ_i has mother σ with state q, left siblings $\sigma_1, \ldots, \sigma_{i-1}$, and right siblings $\sigma_{i+1}, \ldots, \sigma_k$, then σ_i is assigned state q_i. For root nodes, the transition $\sigma \Rightarrow q$ assigns state q to σ. Intuitively, sensing tree automata assign states in a top-down

[4] Sensing tree automata cannot be used to capture tier-string constraints. As noted in Graf and De Santo (2019), sensing tree automata cannot regulate movement steps that aren't restricted by both the specifier island constraint and the adjunct island constraint. For example, a sensing tree automaton cannot ensure that every epp$^+$ is targeted by exactly one epp$^-$. But as we just saw, this is easily enforced over epp-tier strings.

fashion and make the assigned state contingent on the mother's state and the labels of the node, its siblings, and its mother. Since sensing tree automata are deterministic, δ must not contain distinct transitions rules with the same left-hand side. A tree t is accepted by $\mathfrak{A}$ iff every node of t is assigned a state.

With these preliminaries out of the way, it is easy to define the sensing tree automaton $\mathfrak{A}_\sigma$ for some requirement function req. As before, $\mathcal{B}$ is the Boolean representation of an FSA with $m \geq 1$ states that generates $r(\sigma)$, and $Q_\mathcal{B}$ is the result of closing the set of square matrices in $\mathcal{B}$ under Boolean matrix multiplication.

Definition 9. We define $\mathfrak{A}_\sigma := \langle \Sigma, Q_\mathcal{B}, \delta \rangle$. For every state $q \in Q$ and all $1 \leq j \leq k$ and $\sigma, \sigma_1, \ldots, \sigma_k \in Q_\mathcal{B}$, we set

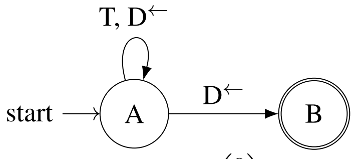
$$q_j := \mathbf{r}_{\sigma_j} \left(q \oplus \bigoplus_{i=1}^{j-1} \mathbf{v}(\sigma_i^{\leftarrow}) \right) \oplus \mathbf{v}(\sigma_j) \qquad \lrcorner$$

such that $\langle q, \sigma(\sigma_1, \ldots, \sigma_j, \ldots, \sigma_k), j\rangle \Rightarrow q_j \in \delta$ iff q_j is defined. Furthermore, $\sigma \Rightarrow \mathbf{v}(\sigma)$ for every $\sigma \in \Sigma$.

The formula for c-strings differs only marginally from T-strings, namely in the argument of $\mathbf{r}$. Quite generally, this is the area where differences between context functions are expressed. To wit, a-strings would also differ in only this area by simplifying the argument of $\mathbf{r}$ to just q. The specific differences between c-strings and T-strings are due to siblings taking on a similar role in c-strings to ancestors in T-strings. With c-strings, an opaque node renders inaccessible all information about its left siblings, and thus the values of siblings have to be combined with $\oplus$ instead of $\otimes$. Also, each sibling σ_i is a c-commander and hence its value must be $\mathbf{v}(\sigma_i^{\leftarrow})$ rather than $\mathbf{v}(\sigma_i)$. These minor changes in the formulas cannot distract from the fact, however, that a-strings, c-strings and T-strings (which includes daughter strings) have remarkably similar automaton implementations.

3.4 Example: Binding and (reduced) c-strings

As with T-strings, we provide a linguistic example of the automaton construction for c-strings. Consider once more the simplified version of Principle A from example 7: if n is a reflexive, then $cs(n)$ must have a non-containing D-head D$^{\leftarrow}$to the right of the rightmost containing T-head.



$$\mathbf{I} := \begin{pmatrix} 1 & 0 \end{pmatrix} \qquad \mathbf{F} := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\mathbf{D}^{\leftarrow} := \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \quad \mathbf{T} := \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \mathbf{id} := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Figure 5: FSA and Boolean matrices for Principle A over reduced c-strings that only contain D$^{\leftarrow}$ and T.

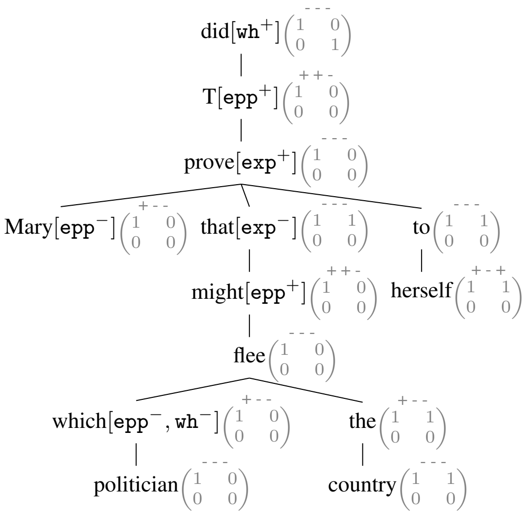


Figure 6: Run of automaton enforcing L_A over reduced c-strings that only contain D and T; $+$ and $-$ indicate whether the node is visible, opaque, and/or restricted (D-heads are visible only when they are non-containing c-commanders, and T-heads are visible only when they are containing c-commanders)

In order to make the example more insightful, we will implement this as a tree automaton that constructs a reduced version of the c-string that only contains Ds and Ts, ignoring all c-commanders that are immaterial to this constraint. That is to say, the task of ignoring irrelevant nodes is shifted from the requirement function into the context function. Hence the requirement function maps reflexives to the string language $L_A := \{\mathrm{T}, \mathrm{D}^{\leftarrow}\}^* \ \mathrm{D}^{\leftarrow}$ (see Fig. 5). The set of visible nodes consists of all D$^{\leftarrow}$ and all T (but not D or T$^{\leftarrow}$). The only opaque nodes are visible T-heads, and the only restricted node is *herself*.

The run of the resulting sensing tree automaton is shown in Fig. 6. Since every node is assigned a state, the tree is correctly recognized as well-formed. The important thing to keep in mind is that each node n may now exhibit a dual behavior depending on whether the formula uses $\mathbf{v}(n)$ or $\mathbf{v}(n^{\leftarrow})$. For example, the state of *Mary* is computed with $\mathbf{v}(\textit{Mary}) = \mathbf{id}_2$, and hence it is identical to the state of its mother *prove*. On the other hand, the state of *that* is computed with $\mathbf{v}(\textit{Mary}^{\leftarrow}) = \mathbf{D}^{\leftarrow}$, inducing a state change. The same effect obtains with the state of *which* relative to *the*. Also note how *might*, by virtue of being opaque, receives the state $\mathbf{v}(\textit{might}) = \mathbf{T}$ and thus renders *Mary* inaccessible from within that subtree.

4 Conclusion

All syntactic constraints that have been put forward in the subregular literature can be analyzed as serial constraints. Serial constraints consists of a context function that associates every node in a tree with a string derived from the tree, and a (regular) requirement function that requires the node's associated string to belong to a (regular) string language. While requirement functions are fairly unremarkable from a formal perspective, context functions require additional considerations.

This paper shows that the context functions for T-strings (and by extension daughter strings) as well as c-strings (and by extension a-strings) can all be implemented as tree automata that follow a universal template. Crucially, the states of these automata store no information beyond what is needed to simulate the requirement function. All other decisions are made based only on the information available directly in each transition rule: whether a node is visible, opaque, and/or restricted. How exactly this information is used to compute states can be succicntly expressesd via matrix multiplication formulas. Each such formula is of the form $q := \mathbf{r}_\sigma\,(\phi) \oplus \mathbf{v}(\sigma)$, where ϕ is a Boolean matrix computed from the states and/or the values of nodes accessible in the transition rule. The general upshot is that even though selectional constraints and constraints on tree tiers seem intuitively different from a-string and c-string constraints, they are but minor variations of a common theme.

One surprising implication of these findings is that (most, perhaps even all) syntactic constraints can be regarded as operating over strings rather than trees. All the regulating work is done by the requiremnt function (which is an FSA), with tree automata serving as a simple wrapper that passes information into this function. Of course this may be due to serial constraints providing a lot more power than it currently seems. For example, even the requirement that a tree must have an odd number of nodes can be implemented as a serial constraint (e.g. via a preorder traversal). On the other hand, it seems that no serial constraint can express the requirement that a tree must contain an even number of nodes that each properly dominate at least two nodes. Further work is needed to properly assess the power of serial constraints and how it varies with the chosen automaton model.

Acknowledgements

The work carried out for this project was supported by the National Science Foundation under Grant No. BCS-1845344. Kenneth Hanson received additional funding from the Institute for Advanced Computational Science at Stony Brook University. We are grateful to three anonymous reviewers for their detailed feedback and suggestions. Reviewer 3 in particular went far beyond the call of duty, and their thoughtful observations will be very useful for future iterations of this work.

References

H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. TommasiK. 2008. Tree automata: Techniques and applications. Published online: http://www.grappa.univ-lille3.fr/tata. Release from November 18, 2008.

Robert Frank and K Vijay-Shanker. 2001. Primitive c-command. *Syntax*, 4(3):164–204.

Doreen Georgi. 2017. Patterns of movement reflexes as the result of the order of merge and agree. *Linguistic Inquiry*, 48:585–626.

Saul Gorn. 1967. Explicit definitions and linguistic dominoes. In *Systems and Computer Science, Proceedings of the Conference held at University of Western Ontario, 1965*, Toronto. University of Toronto Press.

Thomas Graf. 2022a. Diving deeper into subregular syntax. *Theoretical Linguistics*, 48:245–278.

Thomas Graf. 2022b. Subregular linguistics: Bridging theoretical linguistics and formal grammar. *Theoretical Linguistics*, 48:145–184.

Thomas Graf. 2022c. Typological implications of tier-based strictly local movement. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2022*, pages 184–193.

Thomas Graf. 2023. Subregular tree transductions, movement, copies, traces, and the ban on improper movement. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2023*, pages 289–299.

Thomas Graf and Aniello De Santo. 2019. Sensing tree automata as a model of syntactic dependencies. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 12–26, Toronto, Canada. Association for Computational Linguistics.

Thomas Graf and Nazila Shafiei. 2019. C-command dependencies as TSL string constraints. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 205–215.

Ferenc Gécseg and Magnus Steinby. 1997. Tree languages. In Gregorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 1–68. Springer, New York.

Riny Huybregts. 1984. The weak adequacy of context-free phrase structure grammar. In Ger J. de Haan, Mieke Trommelen, and Wim Zonneveld, editors, *Van Periferie naar Kern*, pages 81–99. Foris, Dordrecht.

Gregory M. Kobele. 2006. *Generating Copies: An Investigation into Structural Identity in Language and Grammar*. Ph.D. thesis, UCLA.

Andras Kornai. 1985. Natural language and the Chomsky hierarchy. In *Proceedings of the EACL 1985*, pages 1–7.

Dakotah Lambert and James Rogers. 2020. Tier-based strictly local stringsets: Perspectives from model and automata theory. In *Proceedings of the Society for Computation and Linguistics*, volume 3, pages 330–337.

James McCloskey. 2001. The morphosyntax of wh-extraction in Irish. *Journal of Linguistics*, 37:67–100.

Jens Michaelis and Marcus Kracht. 1997. Semilinearity as a syntactic invariant. In *Logical Aspects of Computational Linguistics*, volume 1328 of *Lecture Notes in Artifical Intelligence*, pages 329–345. Springer.

Uwe Mönnich. 2006. Grammar morphisms. Ms. University of Tübingen.

Frank Morawietz. 2003. *Two-Step Approaches to Natural Language Formalisms*. Walter de Gruyter, Berlin.

Daniel Radzinski. 1991. Chinese number names, tree adjoining languages, and mild context sensitivity. *Computational Linguistics*, 17:277–300.

Nazila Shafiei and Thomas Graf. 2020. The subregular complexity of syntactic islands. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2020*, pages 272–281.

Stuart M. Shieber. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8(3):333–345.

Edward P. Stabler. 1997. Derivational Minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–95. Springer, Berlin.

Edward P. Stabler. 2011. Computational perspectives on Minimalism. In Cedric Boeckx, editor, *Oxford Handbook of Linguistic Minimalism*, pages 617–643. Oxford University Press, Oxford.

A Background: MGs and tree structures

Every MG is fully specified by its *lexicon*, which is a finite set of feature-annotated lexical items that are combined by the structure-building operations Merge and Move. Merge features encode the lexical item's category (*category feature* $\texttt{F}^{-}$) and its subcategorization requirements (*selector features* $\texttt{F}^{+}$). Move features indicate whether the lexical item furnishes any landing sites for movement (*licensor features* $\texttt{f}^{+}$), and whether it undergoes any movement of its own (*licensee features* $\texttt{f}^{-}$). To reduce clutter, we only indicate Move features throughout this paper. Some variants of MGs allow licensor features to indicate whether the landing site is linearized to the left or to the right. We allow this option in this paper and use it for extraposition of the *that*-clause in Fig. 1, but nothing hinges on that.

As is common in subregular syntax, but unlike standard MGs, we assume that licensee features are unordered (this has no effect on generative capacity). For example, a lexical item with both $\texttt{epp}^{-}$ and $\texttt{wh}^{-}$ has to undergo both epp-movement and wh-movement, but the order is unspecified. If the closest epp-landing site is closer than the closest wh-landing site, epp-movmeent will precede wh-movement, otherwise it will follow it. When a lexical item undergoes movement, it does not move by itself but moves along the entire phrase it is the head of.

MG derivations can be represented as dependency trees. The mother-of relation corresponds to Merge steps, and the right-to-left order of siblings matches the order in which they are merged with the mother. For example, *flee* in Fig. 7 first merges with (the phrase headed by) *the*, taking it as a complement. After that, (the phrase headed by) *which* is merged as a specifier. Movement is only indicated by licensor and licensee features — movers are not displaced from their base position. A mover with $\texttt{f}^{-}$ always targets the cloest landing site from its base position that is provided by a matching $\texttt{f}^{+}$.

While MG dependency trees look different from standard phrase structure trees, they encode all necessary syntactic information. However, they do so much more compactly than more common alternatives such as X′-trees (cf. Fig.7).

B Background: FSAs as Boolean matrix multiplication

An FSA is a 5-tuple $A := \langle \Sigma, Q, I, F, \Delta \rangle$ where Σ is the alphabet, Q is a finite set of states, $I \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of final states, and Δ is a finite set of transition rules of the form $q \overset{\sigma}{\Rightarrow} q'$. We assume w.l.o.g. that Q is not empty.

The corresponding **Boolean representation** is constructed as follows. First, we fix an arbitrary enumeration of all $n \geq 1$ states of Q. Then every $\sigma \in \Sigma$ is associated with a $n \times n$ matrix $\mathbf{b}(\sigma)$ such that the cell $\mathbf{b}(\sigma)_{i,j}$ in row i, column j ($1 \leq i, j \leq n$) is 1 if Δ contains the transition $q_i \overset{\sigma}{\Rightarrow} q_j$. Otherwise, the cell is 0. The *initial matrix* $\mathbf{I}$ is a $1 \times n$ matrix such that $\mathbf{I}_{1,j}$ is 1 if $q_j \in I$ and 0 otherwise. Similarly, the *final matrix* $\mathbf{F}$ is a $n \times 1$ matrix such that $\mathbf{F}_{i,1}$ is 1 if $q_i \in F$ and 0 otherwise.

Example 10. The smallest deterministic FSA over $\Sigma := \{a, c\}$ that recognizes $a(aa)^*$ corresponds to the matrices below.

$$\mathbf{I} := \begin{pmatrix} 1 & 0 \end{pmatrix} \qquad \mathbf{F} := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\mathbf{b}(a) := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \mathbf{b}(c) := \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

⌟

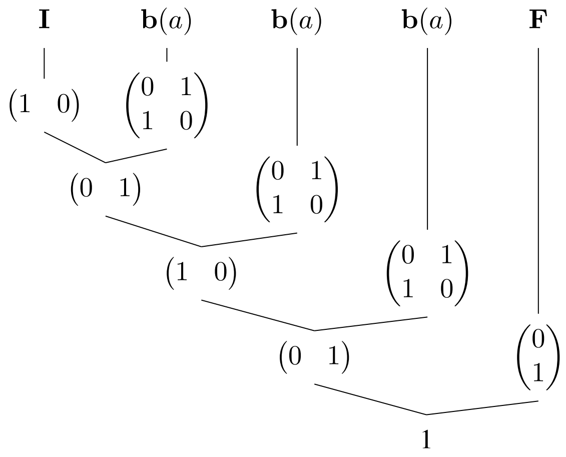
A string $\sigma_1 \cdots \sigma_k$ is recognized by A iff $\mathbf{I} \otimes \mathbf{b}(\sigma_1) \otimes \cdots \otimes \mathbf{b}(\sigma_k) \otimes \mathbf{F} = 1$, where $\otimes$ denotes Boolean matrix multiplication. Given a Boolean $u \times v$ matrix A and $v \times w$ matrix B, $A \otimes B$ is a $u \times w$ matrix C such that for all $1 \leq i \leq u$ and $1 \leq j \leq w$

$$C_{i,j} := \bigvee_{k=1}^{v} (A_{i,k} \wedge B_{k,j})$$

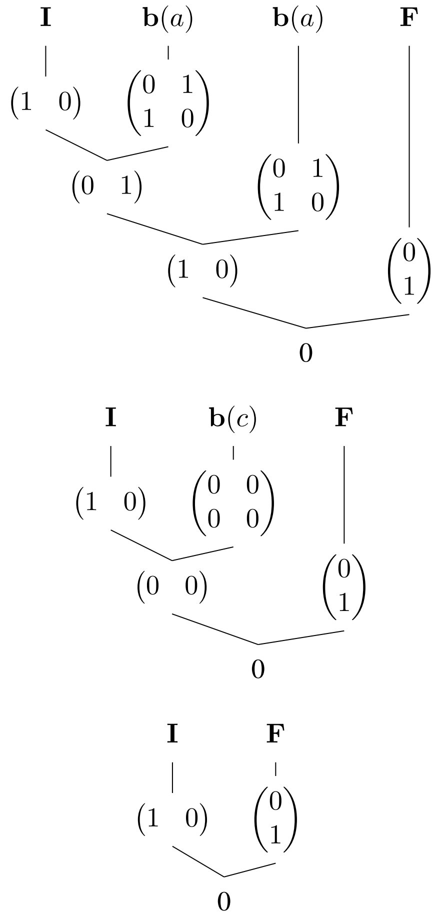
Example 11. Continuing the previous example, the FSA recognizes aaa as we have

$$\mathbf{I} \otimes \mathbf{b}(a) \otimes \mathbf{b}(a) \otimes \mathbf{b}(a) \otimes \mathbf{F} = 1$$

which can be gleaned from the following tree:



But the FSA rejects aa, c, and the empty string:



⌟

The identity matrix $\mathbf{id}_n$ of size n is the $n \times n$ square matrix such that $\mathbf{id}_{i,j} = 1$ if $i = j$, and 0 otherwise. When M is an $m \times n$ matrix, $\mathbf{id}_m \otimes M = M \otimes \mathbf{id}_n = M$.

Example 12. Multiplying $\mathbf{b}(a)$ with $\mathbf{id}_2$ yields

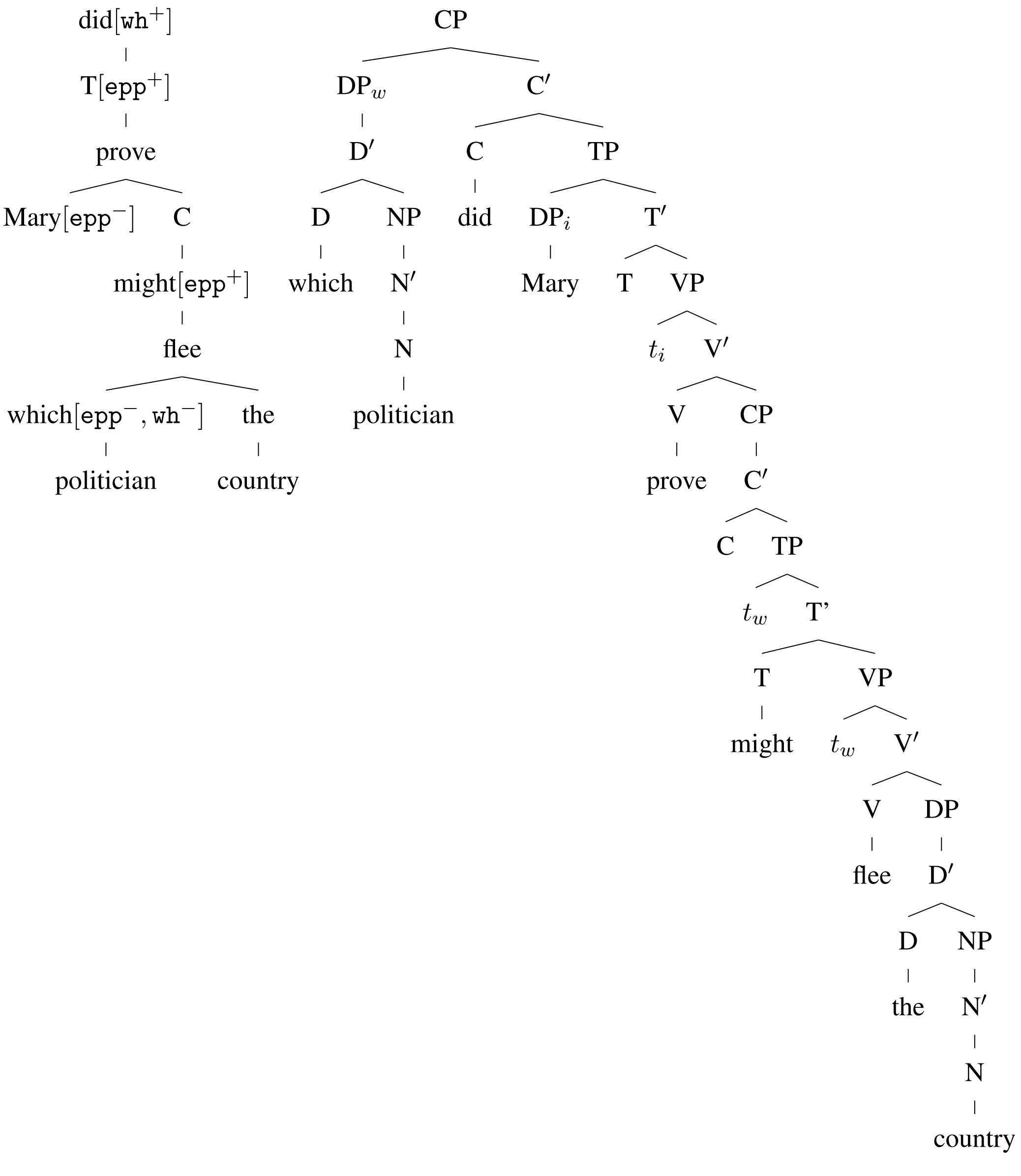


Figure 7: MG dependency tree and corresponding X′-tree for *Which politician did Mary prove might flee the country*

$\mathbf{b}(a)$.

$$\begin{aligned}\mathbf{b}(a) \times \mathbf{id}_2 &= \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix} \otimes \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\\ &= \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\\ &= \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix} \otimes \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\\ &= \mathbf{id}_2 \otimes \mathbf{b}(a) \qquad\qquad \lrcorner\end{aligned}$$

Sparks of Pure Competence in LLMs: the Case of Syntactic Center Embedding in English

Daniel Hardt
Copenhagen Business School
dha.msc@cbs.dk

Abstract

Linguistic theory distinguishes between competence and performance: the competence grammar ascribed to humans is not always clearly observable, because of performance limitations. This raises the possibility that an LLM, if it is not subject to the same performance limitations as humans, might exhibit behavior closer to a pure instantiation of the human competence model. We explore this in the case of syntactic center embedding, where, the competence grammar allows unbounded center embedding, although humans have great difficulty with any level above one. We study this in four LLMs, and we find that the most powerful model, GPT-4, does appear to be approaching pure competence, achieving high accuracy even with 3 or 4 levels of embeddings, in sharp contrast to humans and other LLMs.

> "The heptapods had no objection to the center-embedding of clauses, something that quickly defeated humans".
>
> – Story of Your Life (Chiang, 1998)

1 Introduction

Until recently, there was a simple reason why every AI system would fail the Turing Test – they lacked the basic linguistic capabilities shared by all native speakers of a language. That has changed with current large language models (LLMs), which, it would seem, have now mastered human language. As Mahowald et al. (2024, p. 518) put it, "for modern LLMs, formal [linguistic] competence in English is near human-level". There remain, however, notable differences in the linguistic behavior of LLMs and humans. In this paper we focus on differences in the interpretation of syntactic center embedding constructions. These constructions, while little noted in the NLP literature, have a special significance in the development of modern linguistics. Famously, Chomsky claims that center embedding is fully grammatical as a matter of linguistic competence, but generally fails to be accepted because of a performance limitation involving short-term memory (Chomsky, 1957; Chomsky et al., 1963). These claims are central to the very founding of modern linguistics.

It is revealing to compare center embedding with left and right embedding. Consider a propositional verb like "believe", that can take a sentence as its complement to the right, and that sentential complement might itself involve such a structure, as in (1):

(1) a. [John believes [Harry likes fish]]
b. [John believes [Tom said [everyone knows . . . [Harry likes fish] . . .]]]

An adverbial phrase like "in the library" can modify a verb phrase to its left; the modified verb phrase might itself contain such a modifier, as shown by (2):

(2) a. Col. Mustard [[killed Mr Boddy] in the library]
b. Col. Mustard [[[. . . [killed Mr Boddy] with the candlestick] in the library] . . . without remorse.]

The above cases illustrate the potential for unbounded levels of embedding, both to the right and to the left. We turn now to center embedding. Here the embedding clause contains material both to the left and right of the embedded clause. This is illustrated by (3), where a nominal expression, "teacher", is modified by a relative clause, "the student saw".[1]

(3) [The teacher [the student saw t] is happy.]

[1] The relative clause "the student saw" includes a trace or variable, which we indicate with t to show that it in this case is bound by "the teacher", and similarly with the variables s, d, and g in examples (4) - (6), standing for "student", "driver" and "girl", respectively.

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 346-355.
Eugene, Oregon, July 18-20, 2025

Level 1

Multiple levels of center embedding are readily constructed. Examples (4) - (6) represent levels 2-4 of center embedding.

(4) [The teacher [the student [the driver hit s] saw t] is happy.] **Level 2**

(5) [The teacher [the student [the driver [the girl likes d] hit s] saw t] is happy.] **Level 3**

(6) [The teacher [the student [the driver [the girl [the man hates g] likes d] hit s] saw t] is happy.] **Level 4**

Such multiple center embeddings are generally uninterpretable for human language users, and are virtually nonexistent in normal texts.

In this paper, we explore whether LLMs can interpret and assess center embedding structures in English. We create synthetic data instantiating levels 1-4, and pose questions which require understanding of the structure. For example, for example (4) above, we ask, "Who hit who?", a question that targets the most deeply embedded predication. Here, we find that GPT-4 performs extremely well at all levels, from 1 to 4, in contrast to other models, and also sharply contrasting with what is known about human behavior. This, we argue, suggests that GPT-4 is approaching pure competence. We perform a total of four different tests, varying the embedding level that is questioned, the number of few-shot learning examples provided, and the lengths of NPs in the synthetic data. We also test the ability to assess the grammaticality of center embedding structures.

The results of these additional tests are mixed. On the one hand, in all the tests, there are settings in which GPT-4 performs with very high accuracy, suggesting something close to a pure instantiation of the competence model. On the other hand, there are also tasks and settings in which its performance is degraded, revealing sensitivity to factors such as the embedding level of the question, the number of few-shot examples, and the lengths of the NPs in the structures.

In light of these mixed results, it is premature to conclude that we can observe pure competence in an LLM like GPT-4. Yet its behavior is much closer to pure competence than human behavior. We discuss the implications of this, noting that GPT-4 has attained these impressive abilities, despite the fact that multiple center embeddings are undoubtedly extremely rare in its training data. We conclude with some reflections about the implications of these results for theorizing about the language faculty as it is instantiated in humans as well as in AI models.

2 Related Work

2.1 Center Embedding and Linguistic Competence

According to Karlsson (2007, p. 365) "the mainstream view...voiced by many linguists from different camps" is that "there are no grammatical restrictions on multiple center-embedding of clauses." This is all the more striking given the extreme rarity of multiple center embedding. Karlsson (2007, p. 378) reports on a study of "corpus data from seven Standard Average European (SAE) languages: English, Finnish, French, German, Latin, Swedish, and Danish", finding that "in ordinary language use, written C3s [level 3] and spoken C2s [level 2] are almost non-existent."

Chomsky et al. (1963) present sentence (7), which is an example of level 2 center embedding:

(7) The rat the cat the dog chased killed ate the malt.

In the view of Chomsky et al., example (7) "is surely confusing and improbable but it is perfectly grammatical and has a clear and unambiguous meaning." This argument relies on the Chomskyan distinction between competence and performance, where competence is an idealized theory of the "mental reality underlying actual behavior" (Chomsky, 1965, p. 4). Millière (2024) points out that "Linguistic performance can be affected by external factors like memory limitations, distractions, slips of the tongue, etc. that may obscure the full extent of the underlying competence." Performance factors make the underlying linguistic competence difficult to observe in humans, much as friction makes it difficult to observe the underlying nature of Newton's law of gravity.

2.2 Center Embedding and Performance Factors

Gibson (1998, p. 3) notes that center embedding structures give rise to what is often "referred to as a *processing overload* effect." Gibson proposes the Syntactic Prediction Locality Theory (SPLT). According to this theory, center embedding incurs a memory cost, associated with "computational resources [that] are required to store a partial input sentence" (Gibson (1998, p. 8)). This is an essential feature of center embedding constructions; for example, in (4) above, when the word "driver" is encountered, there are three partial input sentences that must be stored. On this theory, it is the requirement to keep multiple partial structures in memory that can lead to processing overload. Gibson (1998, p. 14) observes that this "... fits with what is known about short-term memory recall in non-linguistic domains: it is harder to retain items in short-term memory as more interfering items are processed."

Gibson considers a wide range of differences in types of embedding structures in arguing for the superiority of SPLT over previous theories, such as Chomsky et al. (1963), Miller and Isard (1964), and Abney and Johnson (1991). What Gibson's theory shares with the previous theories is the view that the facts about center embedding structures are explained with reference to performance factors.

2.3 Human Performance

There are numerous empirical studies that support the claim that center embedding presents difficulties for humans. Thomas (1995, p. 22) asks subjects to rate examples according to perceived difficulty "on a quick first reading". Thomas shows that there are important differences based on the type of center embedding. However, in general, he notes that a simple level 1 structure "is easy to understand", while "embedding just one more clause [i.e. level 2]... produces near incomprehensibility" (Thomas, 1995, p. 8). Bach et al. (1986) describe a psycholinguistic study concerning somewhat different embedding constructions in German and Dutch, again finding a striking difference in difficulty between level 1 and higher levels of embedding. We performed a small, informal survey to further examine human performance on center embedding. See A.2 for details.

2.4 Linguistic Probing of LLMs

There is an extensive literature describing the probing of LLMs for specific linguistic capabilities. Mahowald et al. (2024) argue that current LLMs have largely mastered what they call "formal linguistic competence". They point out that current models perform well on resources such as the BLiMP benchmark (Warstadt et al., 2020), which consists of minimal pairs illustrating many linguistic phenomena. "Models achieve similarly impressive results," they continue, "on other linguistic benchmarks like SyntaxGym" (Gauthier et al., 2020).

However, some recent works have shown that there remain specific capabilities that pose difficulties for some of the most powerful current models. For example Hardt (2023) shows that recent LLMs struggle with the phenomenon of ellipsis while Cui et al. (2023) find that they have substantial difficulties interpreting sentences with "respectively".

2.4.1 Subject-Verb Agreement

A particular area of interest for linguistic probing is subject-verb agreement. Wilson et al. (2023, p. 278) point out that subject-verb agreement "depends not on linear proximity to the verb, but structural proximity ... ", as illustrated by the following paradigm:

(8) a. The labels on the bottle is ...
b. * The label on the bottle is ...
c. * The labels on the bottle are ...
d. The label on the bottle are ...

Humans sometimes diverge from the pure competence model, making errors based on an "attractor", i.e., a noun that intervenes between subject and verb, such as "bottle" in example (8)b above. Recent work (Linzen et al., 2016; Lakretz et al., 2021) has shown that models are able to largely capture the "structure-sensitive grammatical knowledge" implicated in the competence model (Wilson et al., 2023, p. 278), while also showing some errors based on attractor effects.

2.4.2 Center Embedding

Just as with subject-verb agreement, human performance diverges from the competence model with center embedding. However, the divergence is much starker in the case of center embedding – humans consistently fail in the interpretation of multiple center embeddings, although they are completely acceptable according to the competence model. Recent probing of LLMs reveals similar divergence from the competence model. For example Dentella et al. (2023) find that LLM "accuracy on grammatical prompts of center-embedded sentences is at chance" in a test of grammatically judgments by LLMs in the GPT-3 family. Hu et al. (2024, p. 10) test LLMs on a variety of constructions, finding that models "evaluated on the same sentences in minimal pairs achieve at- or near-ceiling performance on most linguistic phenomena tested, except for centre embedding", noting that, for center embedding, “humans also perform near chance.”

An additional observation comes from Gibson and Thomas (1999), concerning what they call the “VP illusion”, where ungrammatical versions of center embedding sentences are judged to be as acceptable as their grammatical counterparts, as illustrated by (9):

(9) a. The patient who the nurse who the clinic had hired met Jack.
b. The patient who the nurse who the clinic had hired admitted met Jack.

Example (9)b is a grammatical level 2 example of center embedding, while (9)a is ungrammatical, since the verb “admitted” is omitted. Gibson and Thomas find that the ungrammatical examples with a missing VP, like (9)b, are judged to be as acceptable as their grammatical counterparts. Subjects were given seven “practice examples”, with “discussion of possible scores for each” (Gibson and Thomas, 1999, p. 238). The study was performed using a questionnaire, and the authors note that, although subjects were instructed to read examples only a single time, subjects had the opportunity to re-read examples. Christiansen (1997) reports on a variant of this study where examples are presented online, so that re-reading is not possible. In this experiment, the missing VP examples were perceived as more acceptable than their grammatical counterparts. See also Engelmann and Vasishth (2009) for an alternative view, arguing that the illusion does not arise for German speakers.

3 Data

We construct a synthetic dataset, where each item consists of a prompt, a context, and a question.[2] We consider each of these elements in turn.

3.1 Context

The context consists of synthetic examples of center embedding of levels 1-4. The form of these examples is as follows, where N denotes Noun, TV denotes Transitive Verb and IV denotes Intransitive Verb:

Level 1: The N the N TV IV.

Level 2: The N the N the N TV TV IV.

Level 3: The N the N the N the N TV TV TV IV.

Level 4: The N the N the N the N the N TV TV TV TV IV.

We have the following substitutions for N and TV:

- **N**: teacher, student, driver, girl, man, woman, boy
- **TV**: saw, hit, likes, hates, knows
- **IV**: is happy, left, is glad

The synthetic data is constructed for levels 1-4, by a procedure that repeatedly makes random selections for N, TV, and IV, resulting in a large collection of sentences for each level. For each test, a random subset of unique sentences are selected.

3.2 Prompt

We define the prompt P0, shown in figure 1. We also use prompts with examples, thus applying few-shot learning. The examples within the prompt are always of the same embedding level as the example in the context.

You will be given an example consisting of a context and a question to answer. The answer should always be of this form "The N V the N", where N stands for a noun, and V stands for a verb.
Context: {context}
Question: {question}
Now answer the question:

Figure 1: Prompt P0

We will use prompts with varying numbers of few-shot examples, such as P5, P10 and P20, i.e., with 5, 10 and 20 few-shot examples respectively.

3.3 Question

We formulate a question, “Who TV’ed who”, where the verb TV is from the most deeply embedded clause. We term this question, Q0 (figure 2).

We also define a question, Q1, that targets the next most deeply embedded predication, as exemplified in figure 3. Note that Q1 does not apply to level 1 examples.

We evaluate the model response as correct if it matches the predefined answer exactly, and incorrect otherwise. All tests use accuracy as the metric.

[2] Data and associated code will be made available on Github upon acceptance.

Level 1
Context: The teacher the student saw is happy.
Q: Who saw who?
A: the student saw the teacher.
Level 2
Context: The teacher the student the driver saw hit is happy.
Q: Who saw who?
A: the driver saw the student.
Level 3
Context: The teacher the student the driver the girl saw hit likes is happy.
Q: Who saw who?
A: the girl saw the driver.
Level 4
Context: The teacher the student the driver the girl the man saw hit likes hates is happy.
Q: Who saw who?
A: the man saw the girl.

Figure 2: Four Embedding Levels with Question Q0, targeting the most deeply embedded structure

4 Testing

4.1 Test 1: Question Q0

For each embedding level (1-4), we test four models: GPT-3.5, GPT-4, llama3-70B and llama3-8B (see Appendix A.1 for details). Test 1 uses question Q0, with either 0 or 5 few-shot examples. In table 1 we present results. GPT-4 is perfect at level 1 with both few-shot settings. With 0 examples, accuracy declines rapidly with higher embedding levels, while with 5 examples, GPT-4 continues to have very high accuracy up to level 4. The other models all have much lower accuracy than GPT-4, especially with higher embedding levels. According to the competence model, center embeddings are fully grammatical at any level. With 5 few-shot examples, GPT-4 seems closely aligned with the competence model, although there is a modest drop in accuracy at levels 3 and 4. The other three models are more similar to humans, in that they have considerable difficulty with any multiple levels of embedding.

4.2 Test 2: Question Q1

Level 2
Context: The teacher the student the driver saw hit is happy
Q: Who hit who?
A: the student hit the teacher.
Level 3
Context: The teacher the student the driver the girl saw hit likes is happy
Q: Who hit who?
A: the driver hit the student.
Level 4
Context: The teacher the student the driver the girl the man saw hit likes hates is happy
Q: Who hit who?
A: the girl hit the driver.

Figure 3: Embedding Levels 2-4 with Question Q1, targeting the next most deeply embedded structure

In Test 2, we pose question Q1, and we use prompts with few-shot examples, ranging from 0 to 30. One interpretation of the test 1 results is that GPT-4 with 5 examples is indeed approaching pure competence with respect to center embedding, with nearly perfect results up to level 3, and still quite high results with level 4, contrasting sharply with humans and the other LLMs. On the other hand, it could be that the behavior of GPT-4 does not actually reflect the competence model involving unbounded structural embedding; there are other conceivable explanations. It could, for example, be employing a simple linear strategy, where it conducts a search to the left of the verb being questioned to locate the subject and object NP's. Consider the example in figure **??**. When posed with the question "Who saw who?", the strategy might be to locate the two NP's immediately to the left of "saw". The first NP encountered is the subject, and the second is the object. This strategy is perhaps facilitated by the fact that all NPs in our synthetic data consist of two words.

By using question Q1, we seek to rule out a linear strategy along the lines given above. Consider the level 2 example in figure 3. To answer the question, "Who hit who?", it is necessary to search left by first skipping over the verb "saw" and the NP "the driver". While this is not inconceivable, it would seem to be more complicated than is the case with question Q0. In test 2 we also experiment with the number of examples in few shot learning, using prompts with up to 30 few-shot examples.

The results are given in table 2. The llama models struggle with Q1, even at level 2. Here GPT-3.5 also struggles, although accuracy does increase markedly as the number of few-shot examples increases. Things are quite different with GPT-4 –

Model	Few-shot	L1	L2	L3	L4
llama3-8b	0	0.005	0.005	0.000	0.000
llama3-8b	5	0.005	0.005	0.005	0.015
llama3-70b	0	0.845	0.640	0.535	0.455
llama3-70b	5	0.760	0.465	0.210	0.095
GPT-3.5	0	0.545	0.355	0.110	0.045
GPT-3.5	5	**1.000**	0.885	0.580	0.315
GPT-4	0	**1.000**	0.500	0.385	0.195
GPT-4	5	**1.000**	**1.000**	**0.900**	**0.845**

Table 1: Test 1 – Question Q0, Accuracy levels 1-4

while it encounters some difficulty with Q1 as compared with Q0, accuracy increases sharply with few-shot examples. Already with 5 examples, GPT-4 is above .9 for levels 2 and 3, and with 25 examples it achieves a score of .840 on level 4.

Model	Few-shot	L2	L3	L4
llama3-8b	0	0.000	0.000	0.000
llama3-8b	5	0.000	0.000	0.000
llama3-8b	10	0.000	0.000	0.000
llama3-8b	20	0.000	0.000	0.000
llama3-70b	0	0.040	0.035	0.040
llama3-70b	5	0.200	0.225	0.010
llama3-70b	10	0.115	0.175	0.130
llama3-70b	20	0.175	0.145	0.000
GPT-3.5	0	0.000	0.000	0.005
GPT-3.5	5	0.565	0.205	0.160
GPT-3.5	10	0.710	0.365	0.075
GPT-3.5	20	0.645	0.325	0.245
GPT-3.5	25	0.870	0.565	0.350
GPT-3.5	30	0.795	0.525	0.315
GPT-4	0	0.165	0.015	0.000
GPT-4	5	0.905	0.980	0.410
GPT-4	10	0.950	0.980	0.335
GPT-4	20	**1.000**	**1.000**	0.435
GPT-4	25	0.995	**1.000**	**0.840**
GPT-4	30	0.995	**1.000**	0.690

Table 2: Test 2 – Question Q1

4.3 Test 3: Variable-Length NPs

In test 3, we create an additional difficulty for the kind of linear strategy discussed above. We modify the test data so that NP's are sometimes two words, and other times three words. This is done by modifying the instantiations for N as follows:

N: happy teacher, young student, driver, girl, man, woman, short boy

Recall that, in our synthetic data, all transitive verbs consist of a single word, and all NP's consist of two words. So, if we consider again the level 2 example in figure 3 with the Q1 question, "Who hit who?" a conceivable search strategy would be: search 4 words to the left, at which point the subject and object NP's are encountered. With variation in the lengths of NPs, a strategy of searching left can no longer be determined by the number of words encountered. Rather, such a strategy would have to be defined in terms of constituents. Results are shown in figure 3. Only GPT-3.5 and GPT-4 are tested here, since the llama models performed so poorly in test 2. It does appear that the variable length of NP's poses an additional challenge for the models. However, similarly to test 2, accuracy rises sharply as few-shot examples increase.

Model	Few-shot	L2	L3	L4
GPT-3.5	0	0.005	0.030	0.015
GPT-3.5	5	0.450	0.270	0.060
GPT-3.5	10	0.710	0.325	0.175
GPT-3.5	15	0.745	0.295	0.090
GPT-3.5	20	0.670	0.285	0.200
GPT-4	0	0.045	0.010	0.005
GPT-4	5	0.995	0.740	0.260
GPT-4	10	0.915	0.830	0.150
GPT-4	15	**0.950**	**1.000**	**0.635**
GPT-4	20	0.870	0.990	0.600

Table 3: Test 3 – Question Q1, variable-length NPs

4.4 Test 4: Missing VP Illusion

In test 4, the model is prompted to judge whether an example is grammatically correct or not. Here we restrict attention to GPT-4. Half of the examples are taken from our original synthetic data, as described above for test 1. We create an equal-sized set of examples with a missing verb, as illustrated for level 2, by (10):

(10) a. *The teacher the student the driver saw is happy.
b. The teacher the student the driver saw hit is happy.

We test with data for levels 2, 3 and 4. The accuracy of judgments is at or below chance (.50) for few-shot values of 0 or 5. However, with few-shot of 10, GPT-4 is performing notably better than humans, well above chance for all three levels. Note that, in the study of Gibson and Thomas (1999), subjects were given 7 “practice examples". Furthermore, they were only tested on level 2 examples.

Model	Few-shot	L2	L3	L4
GPT-4	0	0.405	0.410	0.495
GPT-4	5	0.485	0.525	0.460
GPT-4	10	**0.835**	**0.665**	**0.590**

Table 4: Test 4 – Missing Verb Grammaticality Judgment

4.5 Error Analysis

In all cases, the system is expected to produce answers of the form N1 V N2. We define four types of errors:

- Type 1: N1 is incorrect, N2 is correct
- Type 2: N1 is correct, N2 is incorrect
- Type 3: N1 is incorrect, N2 is incorrect
- Type 4: Other

We consider selected settings based on a manual evaluation of the first 10 examples, restricting attention to GPT-4, in test 1 and test 2. Table 6 shows the number of errors of each type. While there is considerable variation, some clear tendencies can be observed in this small-scale error analysis. With Q0, errors tend to be Type 2, which might relate to the fact that the subject, N1, is adjacent to the verb being questioned. This might explain the comparative lack of errors with N1 for Q0. This is not the case with Q1, and here both type 1 errors and type 3 errors are frequent.

Model	**Level**	**Few-shot**	**Q**	**T1**	**T2**	**T3**
GPT-4	2	0	Q0	0	10	0
GPT-4	3	0	Q0	0	9	1
GPT-4	4	0	Q0	0	9	1
GPT-4	2	5	Q0	10	0	0
GPT-4	3	5	Q0	0	10	0
GPT-4	4	5	Q0	0	10	0
GPT-4	2	0	Q1	0	1	9
GPT-4	3	0	Q1	2	0	8
GPT-4	4	0	Q1	8	0	2
GPT-4	2	5	Q1	10	0	0
GPT-4	3	5	Q1	2	0	2
GPT-4	4	5	Q1	0	7	3

Table 5: Error Types, T1, T2, T3, and T4 for selected test settings, based on manual analysis of first 10 errors for each setting

5 Discussion

Chomsky (1965, p. 4) describes competence as a theory of the “mental reality underlying actual behavior”. As with any domain of natural phenomena, there are an unbounded number of potential theories that are consistent with observation, so other factors, such as elegance and simplicity, play a key role in selecting among candidate theories (Kuhn, 1997). According to the Chomskyan framework, the theory of linguistic competence is formulated in terms of simple recursive rules. While this model sometimes deviates from observed linguistic behavior, these deviations can plausibly be attributed to performance factors.

Dupre (2021, p. 632) notes that, on mainstream views in linguistics, “the gap [between competence and performance] is quite substantial”, and thus finds it unlikely that an LLM would “provide insight . . . to linguistic competence.” Yet this is the conclusion we argue for in this paper – that linguistic competence can be more clearly observed in GPT-4 than in humans.

The evidence for this conclusion has been presented in tests 1-4 described above, and can be largely summarized in figure 4. Here we can see that there are certain settings in which GPT-4 maintains high accuracy in multiple embeddings. In this way GPT-4 differs sharply with the other, less powerful models we tested, and of course this is also quite different from what is observed with human performance.

The evidence we have presented is far from con-

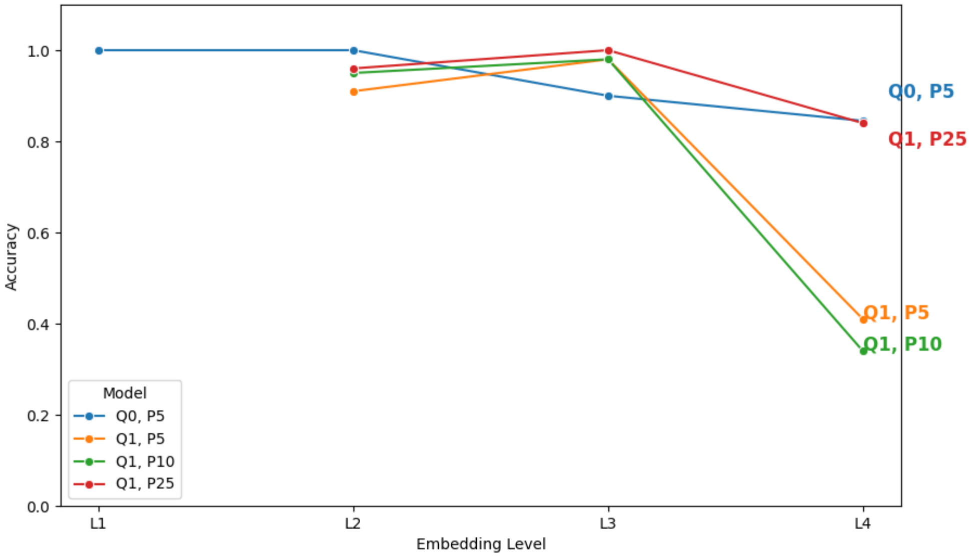


Figure 4: GPT-4 Results, tests 1 and 2
(L1 is not relevant for question Q1)

clusive. Even in the best settings, such as Q0, P5 and Q1, P25, the accuracy is not perfect, and furthermore declines notably at level 4. Our tentative explanation is that, while GPT-4 may well have acquired the linguistic competence model, it is also subject to certain performance limitations, although these limitations are far less severe than those that apply to humans. Another important issue involves few-shot learning. GPT-4 does not achieve high accuracy in the zero-shot setting. It could be argued that GPT-4 does not in fact implement the competence model, but rather, is simply exhibiting effective few-shot learning. We have a different view, based on the idea that it can be difficult to access the knowledge of an LLM through prompt-based tasks. As Hu and Levy (2023, p. 9) argue, "A model's failure to exhibit a linguistic generalization when prompted might not reflect a lack of the relevant information"; Hu and Frank (2024, p. 1) note, furthermore, that "performance on a task is a function of the model's underlying knowledge, combined with the model's ability to interpret and perform the task." We are suggesting that the few-shot learning examples support the model's "ability to interpret and perform the task", thus providing a more accurate reflection of the underlying competence of the model.

6 Conclusions

In this paper, we have explored the possibility that a powerful LLM might reflect pure competence. That is, it might faithfully reflect the human competence model. In humans, linguistic competence is often obscured by performance limitations. Center embeddings present perhaps the most striking divergence between human linguistic behavior and the competence model. We report on a series of tests involving a variety of settings of few-shot learning, embedding levels, and constituent sizes, as well as a grammaticality judgment test. The results are mixed, in that GPT-4 performs very well in many, but not all, settings. We suggest that GPT-4 might be subject to less strict performance limitations than humans, so that competence is less obscured by performance limitations in GPT-4 than it is in humans.

Newton's laws of motion are easier to study in special settings, such as the vacuum chamber of a laboratory. Our hypothesis is that a sufficiently powerful LLM might provide such a frictionless setting in which to observe linguistic competence. While the evidence presented here does not demonstrate that this hypothesis is correct, we hope to have shown that it is worth pursuing, and perhaps it will soon be conclusively demonstrated as LLMs continue to improve.

7 Limitations

The paper seeks to determine whether LLMs understand syntactic center embedding, but this general question is explored in only a few particular ways. Only four LLMs are considered. There are also several important limitations with respect to the data. First, the data is solely English. Second, it is synthetic data, constructed according to a template that reflects one specific form of center embedding, in which a noun phrase is modified by a relative clause. There are other forms of center embedding that could also be considered. In addition, while we have argued that the results are suggestive of a pure competence model, this would of course imply mastery of many other linguistic phenomena, and our investigation has restricted itself to center embedding. Furthermore, while we explored various combinations of different question types, few-shot learning, and constituent lengths, there are other forms and combinations that would be well worth exploring. Finally, we have made claims about the general uninterpretability of multiple center embeddings for humans; while these generally echo claims made in the literature, they are claims that would benefit from rigorous empirical examination.

References

Steven P Abney and Mark Johnson. 1991. Memory requirements and local ambiguities of parsing strategies. *Journal of Psycholinguistic Research*, 20:233–250.

Emmon Bach, Colin Brown, and William Marslen-Wilson. 1986. Crossed and nested dependencies in German and Dutch: A psycholinguistic study. *Language and Cognitive Processes*, 1(4):249–262.

Ted Chiang. 1998. Story of your life. *Stories of your life and others*, pages 117–78.

Noam Chomsky. 1957. *Syntactic structures*. The Hague: Mouton.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. 11. MIT press.

Noam Chomsky, George Armitage Miller, R Luce, R Bush, and E Galanter. 1963. Introduction to the formal analysis of natural languages. *1963*, pages 269–321.

Morten H Christiansen. 1997. The (un) grammaticality of doubly center-embedded sentences: a connectionist perspective. In *Poster presented at the 10th CUNY Sentence Processing Conference, Santa Monica, CA*.

Ruixiang Cui, Seolhwa Lee, Daniel Hershcovich, and Anders Søgaard. 2023. What does the failure to reason with “respectively” in zero/few-shot settings tell us about language models? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8786–8800, Toronto, Canada. Association for Computational Linguistics.

Vittoria Dentella, Fritz Günther, and Evelina Leivada. 2023. Systematic testing of three language models reveals low language accuracy, absence of response stability, and a yes-response bias. *Proceedings of the National Academy of Sciences*, 120(51):e2309583120.

Gabe Dupre. 2021. (what) can deep learning contribute to theoretical linguistics? *Minds and Machines*, 31(4):617–635.

Felix Engelmann and Shravan Vasishth. 2009. Processing grammatical and ungrammatical center embeddings in English and German: A computational model. In *Proceedings of the Ninth International Conference on Cognitive Modeling, Manchester, UK*, pages 240–45.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76, Online. Association for Computational Linguistics.

Edward Gibson. 1998. Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68(1):1–76.

Edward Gibson and James Thomas. 1999. Memory limitations and structural forgetting: The perception of complex ungrammatical sentences as grammatical. *Language and Cognitive Processes*, 14(3):225–248.

Daniel Hardt. 2023. Ellipsis-dependent reasoning: a new challenge for large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 39–47. Association for Computational Linguistics.

Jennifer Hu and Michael C Frank. 2024. Auxiliary task demands mask the capabilities of smaller language models. *arXiv preprint arXiv:2404.02418*.

Jennifer Hu and Roger Levy. 2023. Prompting is not a substitute for probability measurements in large language models. *arXiv preprint arXiv:2305.13264*.

Jennifer Hu, Kyle Mahowald, Gary Lupyan, Anna Ivanova, and Roger Levy. 2024. Language models align with human judgments on key grammatical constructions. *Proceedings of the National Academy of Sciences*, 121(36):e2400917121.

Fred Karlsson. 2007. Constraints on multiple center-embedding of clauses. *Journal of Linguistics*, 43(2):365–392.

Thomas S Kuhn. 1997. *The structure of scientific revolutions*, volume 962. University of Chicago press Chicago.

Yair Lakretz, Dieuwke Hupkes, Alessandra Vergallito, Marco Marelli, Marco Baroni, and Stanislas Dehaene. 2021. Mechanisms for handling nested dependencies in neural-network language models and humans. *Cognition*, 213:104699. Special Issue in Honour of Jacques Mehler, Cognition's founding editor.

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.

Kyle Mahowald, Anna A Ivanova, Idan A Blank, Nancy Kanwisher, Joshua B Tenenbaum, and Evelina Fedorenko. 2024. Dissociating language and thought in large language models. *Trends in Cognitive Sciences*.

George A Miller and Stephen Isard. 1964. Free recall of self-embedded English sentences. *Information and control*, 7(3):292–303.

Raphaël Millière. 2024. Language models as models of language. *arXiv preprint arXiv:2408.07144*.

James David Thomas. 1995. *Center-embedding and self-embedding in human language processing*. Ph.D. thesis, Massachusetts Institute of Technology.

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.

Michael A Wilson, Zhenghao Zhou, and Robert Frank. 2023. Subject-verb agreement with seq2seq transformers: Bigger is better, but still not best. *Proceedings of the Society for Computation in Linguistics*, 6(1):278–288.

A Appendix

A.1 Test Details

A.1.1 Test 1

The GPT-4 and GPT-3.5-turbo models were accessed from the OpenAI site in the period from 13 to 14 December 2024, with default settings. The llama3-70b and llama3-8b models were accessed from api.llama-api.com in the same period, also with default settings. Each of these tests were performed with 200 randomly selected examples.

A.1.2 Test 2

The GPT-4 and GPT-3.5-turbo models were accessed from the OpenAI site in the period from 10 November 2024 to 1 December 2024, with default settings. The llama3-70b and llama3-8b models were accessed from api.llama-api.com in the same period, also with default settings. Each of these tests were performed with 200 randomly selected examples.

A.1.3 Test 3

The GPT-4 and GPT-3.5-turbo models were accessed from the OpenAI site in the period from 10 November 2024 to 1 December 2024, with default settings. Each of these tests were performed with 200 randomly selected examples.

A.1.4 Test 4

The GPT-4 model was accessed from the OpenAI site in the period from 10 November 2024 to 1 December 2024, with default settings. Each of these tests were performed with 200 randomly selected examples.

A.2 Human Performance

We posed 4 examples each of levels 1, 2 and 3, to 9 respondents, for a total of 108 observations. The context and question were modeled after the materials used in our LLM experiments.[3] As shown in table 6 the results show a sharp drop in accuracy from level 1 to levels 2 and 3; consistent with widely held views in the literature.

Level	Accuracy
1	.889
2	.611
3	.528

Table 6: Survey Results for Center Embeddings

[3] Survey data provided online upon acceptance.

Designing a Digital Keyboard for Itunyoso Triqui

Kayla Shames
University at Buffalo
kmshames@buffalo.edu

Abstract

In this paper, I outline the process of creating a mobile keyboard for Itunyoso Triqui, an endangered indigenous language of Southern Mexico. Literacy in Itunyoso Triqui is low, and speakers do most of their texting and typing in Spanish. The language's complex lexical tone system and many multigraphs make typing on a typical QWERTY mobile keyboard difficult. This keyboard offers several innovative features to make typing in Triqui more idiomatic, including multigraph keys and several options for tone input. By allowing for more convenient typing in Triqui, this keyboard enables speakers to use Triqui over Spanish in day-to-day typing and texting, which helps bolster language vitality and literacy. The case of Triqui is not unique. Many minority language speakers use a dominant language for typing and texting; dominant languages have better language tools, due to having a larger potential userbase and therefore more resources for development. By creating tools for minority languages that are idiomatic to each language and convenient to use, we can encourage their use in digital contexts, increasing literacy and vitality of minority languages more broadly.

1 Introduction

As technology use increases and more and more communication is done online through text, there is an increased demand for digital tools that serve endangered language communities. In this paper, I discuss the process of developing a mobile keyboard for Itunyoso Triqui, a minority language spoken in Southern Mexico.

Itunyoso Triqui is one of three Triqui varieties, spoken by roughly 2500 speakers in and around San Martín Itunyoso, Oaxaca, Mexico (DiCanio, 2010). Its orthography uses a modified Latin script, with many multigraphs and accent marks to represent its complex tone system.

Fluent literacy in Itunyoso Triqui is low (1-2%). There have been recent efforts to establish literacy programs, but they are currently on hiatus due to political and economic conditions in the area. Many speakers use at least some digital technology (usually a smartphone), and day-to-day writing and texting is usually done in Spanish, due to low literacy and the relative difficulty of texting in the language.

In creating a digital keyboard for Itunyoso Triqui, we aim to help speakers to practice literacy and use the language in everyday texting and writing. This project is inspired by the TZ'IB'MA project (Mateo Toledo 2022), which created mobile keyboards for several Mayan languages, including Q'anjob'al, Kaqchikel, and Mam, which helped speakers text more easily in their languages.

2 Digital Tools for Minority Languages

Modern language technology development exists in a feedback loop. Because dominant languages have a larger userbase, language technology more likely to be developed for them. Subsequently, bilinguals gravitate toward dominant languages in texting due to their higher-quality, more convenient tools (van Esch et al., 2019). Creating tools for minority languages allows to use their language in text as well as speech, helping both literacy and language vitality.

Texting is an essential part of modern literacy; for many people, the majority of their day-to-day writing is over text. A well-designed keyboard that is idiomatic to a language is helpful for both literacy and language revitalization programs, as it

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 356-358.
Eugene, Oregon, July 18-20, 2025

allows students to practice the written language in day-to-day, spontaneous conversation.

Creating tools for marginalized languages also helps to chip away at the feedback loop, by creating an existing user base for these languages. While private companies currently have little economic incentive to make these tools, academic projects can help fill that gap and seed further development. Digital language technology is another way that linguists can contribute materially to the communities of languages they document.

In designing language tools, one should make them convenient and idiomatic to a language, in order for speakers to want to use them. The desire to use one's native language can only overcome so much frustration with poorly made tools. Therefore, a language's orthography and linguistic structure should be taken into account when making language tools.

3 Keyboard Development

In creating this keyboard, there were several requirements aimed at making the keyboard easy for speakers to adopt. First, the keyboard should be easy to access and install, without advanced technical knowledge. Second, the keyboard should be beginner-friendly; it should be easy and intuitive to use, even without technical knowledge, and it should be usable even without perfect literacy, as part of its use case is in teaching literacy. Lastly, it should be convenient and idiomatic to Triqui; the multigraphs and accent marks that are easy to input.

The keyboard was made using Keyman by SIL, a program for creation of custom keyboards. To install, users download the Keyman mobile app from the App Store or Google Play Store. (The keyboard is only designed for mobile use, as mobile phones are much more common than computers in the community). Users can install the keyboard within the Keyman app, without a technical installation process, and the keyboard works system-wide. The keyboard also includes a predictive text feature, trained on corpus data.

Itunyoso Triqui has a highly complex tone system, contrasting five level tones and four contour tones. The five level tones /a^{5} a^{4} a^{3} a^{2} a^{1}/ are written with single vowels <á á a à à>, and the four contour tones /a^{43} a^{32} a^{31} a^{13}/ are written with double vowels <áa aa aà àa> (Note that the pairs /a^{5} a^{4}/ and /a^{1} a^{2}/ are not distinguished in the orthography). Because of the high functional load of tone, nearly every word has one or more accent marks, and accents cannot be omitted without sacrificing comprehensibility.

A standard QWERTY mobile keyboard requires long-presses to input accents. This keyboard offers long-press accent input as well as several other options that are more idiomatic. These include deadkeys, swiping up and down on vowel keys, a predictive text feature, and a novel method for inputting and correcting accents on already-typed words. We expect the predictive text to be the main input method for accents, and the accent correction feature allows for manually adding accents to words that the predictive text could not predict correctly, without having to delete the entire word.

Many common segments are written with digraphs, with <kw nd ngw ngw ts ch chr cn> representing /kʷ ⁿd ᵑg ᵑgʷ t͡s t͡ʃ ʈ͡ʂ ᶜɲ/, respectively. This keyboard includes keys that input a multigraph in its entirety, replacing keys in the QWERTY layout for letters that are not used in Triqui (e.g. <kw> replacing <w>, <ch> replacing <c>). This phonemic layout with multigraph keys was inspired by a similar approach used in the TZ'IB'MA project (Mateo Toledo 2022).

In addition, most consonants can be geminated, indicated with a double letter. While this is straightforward to type for unigraph consonants (e.g. /k~kː, β~βː/ <k~kk, b~bb>), this can be cumbersome when the singleton consonant is already a multigraph (<kw~kkw, ch~cch, chr~cchr>). This keyboard has a text replacement feature that allows for typing of any geminate with a double tap, including multigraphs (e.g. <kw><kw> to type <kkw>).

4 Distribution Efforts and Future Goals

As of May 2025, the keyboard has 272 downloads, representing roughly 11% of Itunyoso Triqui speakers. In the future, the keyboard will also be incorporated into the literacy program when it resumes, and continually updated based on user feedback.

By designing the keyboard around the orthography and linguistic structure of Itunyoso Triqui, and giving users multiple options to find what works best for them, this project aims to make typing in Triqui as convenient as typing in Spanish, even with limited resources and funding. We hope that the availability of this keyboard will increase the use of Triqui in day-to-day texting and writing, improving literacy and language vitality.

References

DiCanio, C. and Cruz Martínez, B. (2010). *Chungwì Snáhánj nìh, El Mundo Triqui: Palabras de San Martín Itunyoso*. Instituto Nacional de Lenguas Indígenas.

DiCanio, C. (2010). Illustrations of the IPA: San Martín Itunyoso Trique. *Journal of the International Phonetic Association*, 40(2):227–238.

van Esch, D., Sarbar, E., Lucassen, T., O'Brien, J., Breiner, T., Prasad, M., Crew, E., Nguyen, C., & Beaufays, F. (2019). *Writing Across the World's Languages: Deep Internationalization for Gboard, the Google Keyboard* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.1912.01218

Keyman (Version 17). (n.d.). [Computer software]. SIL International. https://keyman.com/

Mateo Toledo, E. (2022). *TZ'IB'MA (Teclado digital para lenguas mayas)* [Computer software]. https://www.lenguasmayas.com/

Do LLMs Disambiguate Italian Relative Clause Attachment?

Michael Kamerath and **Aniello De Santo**
University of Utah
{michael.kamerath,aniello.desanto}@utah.edu

1 Introduction

An important component of human sentence comprehension is how to judge multiple simultaneously correct interpretations for a single input. For instance, consider the case (as in 1) of a relative clause (RC) *that was running* following a complex noun phrase *son of the doctor*:

(1) I saw the son of the doctor that was running.

There are two possible interpretations of this sentence based on whether the RC *that was running* modifies *the son* or *the doctor*. The interpretation in which the RC modifies *the doctor* is referred to as low attachment (LA) while the case of the RC modifying *the son* is referred to as high attachment (HA).

RC attachment preferences thus present an interesting way of probing LLMs' syntactic knowledge, given that speakers' preferences for HA or LA vary cross-linguistically, and has been reported to be affected by a variety of syntactic and semantic factors (Grillo and Costa, 2014). However, RC attachment preferences seem to be understudied in the LLM syntactic evaluation literature (Davis and Van Schijndel, 2020; Issa and Atouf, 2024). Here, we aim to add to this scarce literature, and evaluate a variety of LLMs trained to determine their disambiguation strategies over relative clauses in Italian.

2 Italian RC Attachment

Modulo other variables, English speakers generally exhibit a low attachment RC preference with Italian speakers preferring a high attachment interpretation. Recently, it has been argued that one important predictor of attachment preference in Italian RCs is whether the verb in the main clause is non-perceptual (*marry, know, cook, etc*) or perceptual (*observe, hear, smell, etc*). When other semantic and syntactic aspects are controlled for, RCs of sentences containing non-perceptual verbs lead to a LA preference while perceptual verbs lead to a HA preference (Grillo and Costa, 2014; Lee and De Santo, 2024). Focusing on French, Hénot-Mortier (2023) has shown that monolingual and multilingual transformer architectures exhibit some sensitivity to non-perceptual/perceptual verb type modulations in non-RC contexts. Building on the psycholinguistics literature and these past LLM results on RC attachment and verb type effects, here we ask:

1. whether LLMs tested on Italian show any type of attachment preference;
2. whether these preferences conform to those of Italian speakers;
3. whether these preferences show sensitivity to verb type.

Sentence	Verb Type	Attachment
a	perceptual (P)	HA
b	perceptual (P)	LA
c	non-perceptual (N)	HA
d	non-perceptual (N)	LA

Table 1: Summary of Italian Stimuli by Group

3 Experiment and Results

We build on Grillo and Costa (2014), testing Italian sentences with a structure as in (1) but with a matrix verb manipulation, which is either perceptual or non-perceptual in a 2×2 design summarized in Table 1. As a proxy for a model's attachment preferences, we measure the surprisal value at the embedded verb (Davis and Van Schijndel, 2020). Therefore, we depart from Grillo and Costa (2014) in being unable to use fully ambiguous relative clauses. Instead, we leverage the stimuli used in Lee and De Santo (2024), testing our models on sentences that are disambiguated for HA or LA by gender agreement between one of the two nouns in the complex DP (*son* or *doctors*) and the embedded verb. We can then think of contrasting sentence types pairwise, which leads to the following LA/HA predictions:

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 359-360.
Eugene, Oregon, July 18-20, 2025

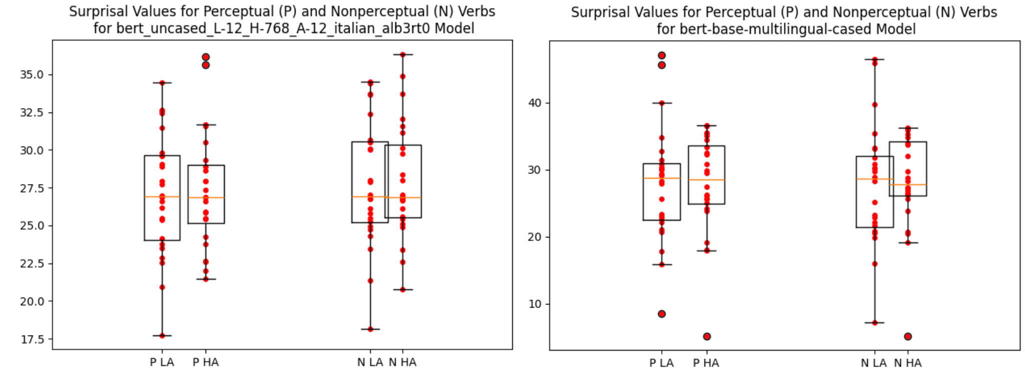


Figure 1: Surprisal Values by Attachment Type and Verb Type for one of the Italian-only (GePpeTto; Polignano et al., 2019) and one of the multilingual (bert-base-079 multilingual-cased; Devlin et al., 2019) models tested.

`Attachment Preference` ← `LOW` if Verb Surprisal(a) $>$ Verb Surprisal(b)
`Attachment Preference` ← `HIGH` if Verb Surprisal(a) $<$ Verb Surprisal(b)
`Attachment Preference` ← `LOW` if Verb Surprisal(c) $>$ Verb Surprisal(d)
`Attachment Preference` ← `HIGH` if Verb Surprisal(c) $<$ Verb Surprisal(d)

We evaluated two Italian-only models, and three multilingual models (GePpeTto; AlBerto; bert-base-multilingual-cased; bert-base-multilingual-cased; xlm-roberta-large) in line with those tested for French by Hénot-Mortier (2023). For each LLM, pairwise contrasts do not reveal a strong tendency towards either LA or HA. We then fit linear mixed-effect models using Surprisal at the embedded verb as the dependent variable, and Verb Type and Attachment Type as fixed effects. Our analyses show no significant attachment or verb type effects, again consistent with the absence of attachment preferences (in line with Italian speakers or not) in each of the models, (see Figure 1 for results for two of the models).

4 Discussion and Further Work

In this work, we measured the difference in surprisal of locally ambiguous sentences at the point of disambiguation to determine whether a variety of LLMs learn human-like attachment preferences in Italian. Our results indicate that none of the models we tested exhibits any attachment preference at all, somewhat in contrast to previous results for Spanish and Arabic (Davis and Van Schijndel, 2020; Issa and Atouf, 2024). However, Davis and Van Schijndel (2020) tested models with an LSTM architecture, while Issa and Atouf (2024) used prompting methods as opposed to the surprisal measurements used here. Future work should explore these differences more in depth, and suggest a primary role for RC disambiguation in the study of LLMs' capabilities cross-linguistically.

References

Forrest Davis and Marten Van Schijndel. 2020. Recurrent neural network language models always learn english-like relative clause attachment. *arXiv preprint arXiv:2005.00165*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Nino Grillo and João Costa. 2014. A novel argument for the universality of parsing principles. *Cognition*, 133(1):156–187.

Adèle Hénot-Mortier. 2023. Do language models discriminate between relatives and pseudorelatives? In *Proceedings of the 2023 CLASP Conference on Learning with Small Data (LSD)*, pages 55–61.

Elsayed Issa and Noureddine Atouf. 2024. Context-biased vs. structure-biased disambiguation of relative clauses in large language models. *Procedia Computer Science*, 244:425–431. 6th International Conference on AI in Computational Linguistics.

So Young Lee and Aniello De Santo. 2024. Online evidence for pseudo-relative effects on italian rc attachment resolution. *Language, Cognition and Neuroscience*, 39(9):1212–1229.

Marco Polignano, Pierpaolo Basile, Marco De Gemmis, Giovanni Semeraro, Valerio Basile, et al. 2019. Alberto: Italian bert language understanding model for nlp challenging tasks based on tweets. In *CEUR workshop proceedings*, volume 2481, pages 1–6. CEUR.

Explaining differences between phonotactic learning biases in the lab and typological trends using Probabilistic Feature Attention

Brandon Prickett
University of Massachusetts Amherst
bprickett@umass.edu

1 Introduction

A primary goal of linguistic theory is to explain why certain kinds of languages are underattested. One methodology that has had success in explaining phonological typology has been artificial language learning, in which participants are trained for a short period of time on a synthetic language that was designed to test the learnability of a particular kind of pattern (for a review of this literature, see Moreton and Pater, 2012a,b). Often, the goal of these experiments is to see if participants' learning biases in the lab might explain typology by showing that underattested languages are more difficult to acquire (see, e.g., Wilson, 2006; Finley, 2008; Glewwe, 2019).

However, learning biases seen in an experimental setting do not always match typological trends. Moreton and Pertsova (2014) implemented a set of patterns introduced by Shepard et al. (1961, henceforth, *Shepard Types*) as phonotactic restrictions and taught them to participants in an artificial language learning experiment. They found that participants' preferred patterns failed to mirror typological trends in a database of attested phonological generalizations (Mielke, 2008).

Here, I model the acquisition of phonotactic patterns that align with the six Shepard Types tested by Moreton and Pertsova (2014) using a maximum entropy phonotactic grammar (Hayes and Wilson, 2008; Moreton et al., 2017) equipped with Probabilistic Feature Attention (Prickett, 2023). This model predicts the biases seen in Moreton and Pertsova (2014)'s experimental results early in learning, but by the end of learning reflects the trends present in phonological typology. These results could help explain the differences observed by Moreton and Pertsova (2014) between artificial language learning and typology, since the latter could be shaped by more long-term learning biases.

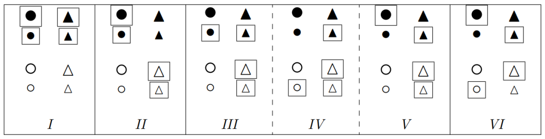


Figure 1: Shepard Type examples using the features [±black], [±circle], and [±large]. Boxes around shapes show how stimuli could be divided up in each type. Taken from Moreton et al. (2017).

2 Background

2.1 Shepard Types

Shepard et al. (1961) found that humans were biased toward certain kinds of patterns when learning in an experimental setting. They used patterns involving 8 stimuli, where each stimulus could be uniquely identified with three features. The shapes in Figure 1 show an example of such a stimulus space. They found that out of the six possible ways of dividing up the space into two equally sized groups, their participants learned some divisions more quickly than others. The roman numerals in Figure 1 show the relative ease with which each type was learned in their original experiments (with lower numbers applied to easier Types and dotted lines between Types representing inconsistent/marginal differences in learnability). For a review of the literature on Shepard Types for nonlinguistic patterns, see Kurtz et al. (2013).

Moreton and Pertsova (2014) implemented the Shepard Types as phonotactic patterns (where the three features were phonological and the stimuli were words). Their results showed that in this context, the Shepard Types were learned in the order (from easiest to most difficult): I, IV, III, V, II, and VI. However, when Moreton and Pertsova (2014) analyzed a database of phonological patterns, assigning as many patterns as they could to each of the Shepard Types, they found that the typological frequency of the Types roughly mirrored the origi-

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 361-364.
Eugene, Oregon, July 18-20, 2025

Segment	[labial]	[continuant]	[voice]
p	+	-	-
b	+	-	+
f	+	+	-
v	+	+	+
t	-	-	-
d	-	-	+
s	-	+	-
z	-	+	+

Table 1: Features and segments used for all simulations presented here.

nal ordering found by Shepard et al. (1961): I, II, III, IV/V, and VI.

2.2 Probabilistic Feature Attention

Prickett (2023) proposed Probabilistic Feature Attention (henceforth, *PFA*) as a way to model certain kinds of uncertainty that likely exist in the process of phonological acquisition. PFA introduces noise into a learning model's training data by making certain segments temporarily ambiguous with one another and is based on a regularization technique from the machine learning literature called *dropout* (Srivastava et al., 2014). This ambiguity is based on the features used to represent the segments, with the model distributing its attention (Nosofsky, 1986) to these features probabilistically and resampling which features are attended to on each learning update.[1]

For example, imagine a phonotactic pattern using the segments and features in Table 1. If the model attended to the feature [continuant], but not [voice], the difference between [t] and [s] would be preserved, but the model would treat [t] and [d] identically. If the model was learning a pattern in which voiceless sounds were grammatical and voiced sounds were not, any learning update in which [voice] was not attended to would fail to push the learner in the correct direction.

Prickett (2023) paired PFA with a maximum entropy phonotactic learner (Hayes and Wilson, 2008) with a conjunctive constraint schema (Moreton et al., 2017) and successfully modeled a number of artificial language learning experiments. Those results demonstrated that some relevant features being attended to while others are not can push the model to generalize and learn in unexpected ways. This altered learning and generalization mirrored the human behavior in the relevant experiments.

3 Methods

The results presented here were found using the software published in the supplementary materials included with Prickett (2023), which implements a maximum entropy phonotactic grammar and trains it with batch gradient descent paired with PFA. The hyperparameter values that were used for these results were a learning rate of .05 and an attention probability of .25. These were chosen after a short amount of piloting, with a full grid search of these values being left to future work.

Constraints representing every possible combination of the features in Table 1 were used (following Moreton et al., 2017). This included constraints with a single valued feature (e.g., *[+voice]), constraints with two valued features (e.g., *[+voice, +continuant]), and constraints with three valued features (e.g., *[+voice, +continuant, -labial]). Constraints with a single feature were always violated by half of the possible segments (e.g., [b, v, d, z]), constraints with two features were always violated by two segments (e.g., [v, z]), and constraints with three features were always violated by a single segment (e.g., [z]).

Six 'languages' (one for each Shepard Type) were implemented using 'words' that were only a single segment long. In the training data for each language, four of the words had a probability of 1 and four had a probability of 0 (representing grammatical and ungrammatical words, respectively). The model was tested in 30 separate runs for each language, since PFA introduces variability into the learning process. This ensured that results were representative of the model's average behavior, and not the random choice of feature attention in a single run.

4 Results

Figure 2 shows the average accuracy for the model with PFA on each pattern. The model's initial ordering of Shepard Types matches the performance observed by Moreton and Pertsova (2014) in their experiment: I, IV, III, V, II, and VI. However, later in learning, the ordering of the patterns mirrors the typological trends observed by Moreton and Pertsova (2014), instead, with Type II crucially having a higher accuracy than III, IV, or V. Note

[1] Note that this ambiguity could arise from a number of factors in real phonological acquisition, such as misperception (Bailey and Hahn, 2005) or constraints on memory (Gathercole and Adams, 1993).

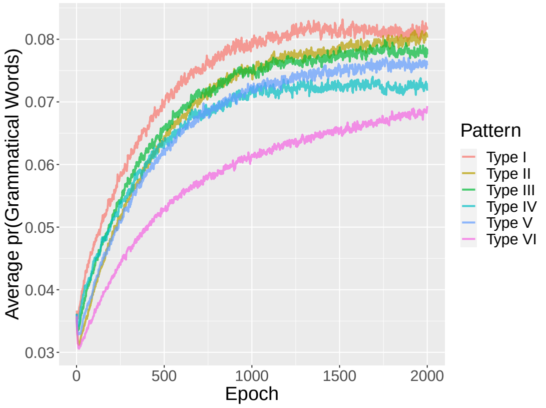


Figure 2: Probability of grammatical words in each pattern, according to the model at each epoch. Results are averaged over 30 separate runs per pattern.

that the ordering of Types IV and V does change toward the end of acquisition, but the relative ordering of these Types in the typological study was also inconsistent.

5 Discussion

5.1 Why does the model capture these biases?

The relative ordering of types that is present early on in the model's learning matches the expected behavir of this kind of maximum entropy learner. The reason that these biases exist in the model is because of the structure of its constraint set and the nature of gradient-based learning algorithms. For an in-depth explanation for how a conjunctive constraint schema combined with gradient descent predicts this ordering of Shepard Types, see Moreton et al. (2017).

But why does PFA cause the model to change its relative ordering of Shepard Types later in learning? The more features that are relevant to a pattern, the more opportunities PFA has to obscure that pattern over the course of learning (for more on this effect, see Prickett, 2023, §4.3). Type II patterns only involve two features, while Types IV and V both involve three. For Types IV and V, all three features must be attended to for a learning update to push the model in the correct direction. But in Type II, only the two relevant features have to be attended to for the model to move its weights in the correct direction. This effect of PFA compounds as learning continues, making IV and V ultimately more difficult to learn.

5.2 Future Work

The relationship between phonological learning in the lab and phonological typology in the real world is still largely an open question. Many factors could drive differences between biases seen in artificial language learning and real-world typology, such as the effect of sleep on acquisition (see e.g., St Clair and Monaghan, 2008), the pressures caused by the iterative and interactive nature of language learning (see e.g., Hughto, 2020), and phonetically driven channel bias (see e.g., Ohala, 2014). The results presented here offer an explanation for one particular mismatch between observed learning biases and the frequency of attested patterns, but future work should explore how PFA might interact with these other phenomena.

Future work should also explore whether other models of phonological learning can explain the results in Moreton and Pertsova (2014). A maximum entropy model that uses a conjunctive constraint schema will always predict the ordering of Shepard Types seen in Moreton and Pertsova (2014)'s experiment unless additional mechanisms are added to it. But other approaches to phonotactic learning, such as induced constraints (see, e.g. Hayes and Wilson, 2008), expectation-driven learning algorithms (Jarosz, 2015), or neural networks (see, e.g. Mayer and Nelson, 2020) could all be tested on these same patterns.

More typological work could also illuminate future directions for this kind of research. Moreton and Pertsova (2014) used patterns across two segments in their experiment, but only had access to single-segment patterns in the database they used to calculate typological frequencies (Mielke, 2008). The simulations presented here used single-segment patterns as well, but PFA can be used with multi-segment sequences (Prickett, 2023) and if future work found a different typological distribution for patterns involving two segments, testing the model on that kind of pattern could be useful.

5.3 Conclusions

While the goal of artificial language learning is usually to explain some kind of typological trend, Moreton and Pertsova (2014) found distinct differences between learning observed in the lab and the frequency of certain patterns in phonological typology. A model with PFA, an independently motivated mechanism (Prickett, 2023), matches Moreton and Pertsova (2014)'s experimental results early in learning, but mirrors typological trends later in acquisition, providing a potential explanation for the mismatch observed by Moreton and Pertsova (2014).

References

Todd M. Bailey and Ulrike Hahn. 2005. Phoneme similarity and confusability. *Journal of Memory and Language*, 52(3):339–362.

Sara Finley. 2008. *Formal and Cognitive Restrictions on Vowel Harmony*. PhD Thesis, Johns Hopkins University.

Susan E. Gathercole and Anne-Marie Adams. 1993. Phonological working memory in very young children. *Developmental Psychology*, 29(4):770–778. Place: US Publisher: American Psychological Association.

Eleanor Glewwe. 2019. *Bias in Phonotactic Learning: Experimental Studies of Phonotactic Implicationals*. PhD Thesis, UCLA.

Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic inquiry*, 39(3):379–440.

Coral Hughto. 2020. *Emergent typological effects of agent-based learning models in maximum entropy grammar*. Ph.D. thesis, University of Massachusetts Amherst.

Gaja Jarosz. 2015. Expectation driven learning of phonology. *Ms., University of Massachusetts Amherst*.

Kenneth J Kurtz, Kimery R Levering, Roger D Stanton, Joshua Romero, and Steven N Morris. 2013. Human learning of elemental category structures: revising the classic result of shepard, hovland, and jenkins (1961). *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 39(2):552.

Connor Mayer and Max Nelson. 2020. Phonotactic learning with neural language models. *Proceedings of the Society for Computation in Linguistics*, 3(1):149–159.

Jeff Mielke. 2008. *The emergence of distinctive features*. Oxford University Press.

Elliott Moreton and Joe Pater. 2012a. Structure and Substance in Artificial-phonology Learning, Part I: Structure. *Language and Linguistics Compass*, 6(11):686–701.

Elliott Moreton and Joe Pater. 2012b. Structure and substance in artificial-phonology learning, part II: Substance. *Language and linguistics compass*, 6(11):702–718.

Elliott Moreton, Joe Pater, and Katya Pertsova. 2017. Phonological Concept Learning. *Cognitive science*, 41(1):4–69.

Elliott Moreton and Katya Pertsova. 2014. Pastry phonotactics: Is phonological learning special. In *Proceedings of the 43rd Annual Meeting of the Northeast Linguistic Society, City University of New York*, pages 1–14. Graduate Linguistics Students' Association Amherst, MA.

Robert M. Nosofsky. 1986. Attention, similarity, and the identification–categorization relationship. *Journal of experimental psychology: General*, 115(1):39.

John Ohala. 2014. The phonetics of sound change. In *Historical linguistics*, pages 237–278. Routledge.

Brandon Prickett. 2023. Probabilistic feature attention as an alternative to variables in phonotactic learning. *Linguistic Inquiry*, 54(2):219–249.

Roger N. Shepard, Carl I. Hovland, and Herbert M. Jenkins. 1961. Learning and memorization of classifications. *Psychological monographs: General and applied*, 75(13):1.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Michelle C St Clair and Padraic Monaghan. 2008. Language abstraction: Consolidation of language structure during sleep. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 30.

Colin Wilson. 2006. Learning phonology with substantive bias: An experimental and computational study of velar palatalization. *Cognitive science*, 30(5):945–982.

`semantic-features`: A User-Friendly Tool for Studying Contextual Word Embeddings in Interpretable Semantic Spaces

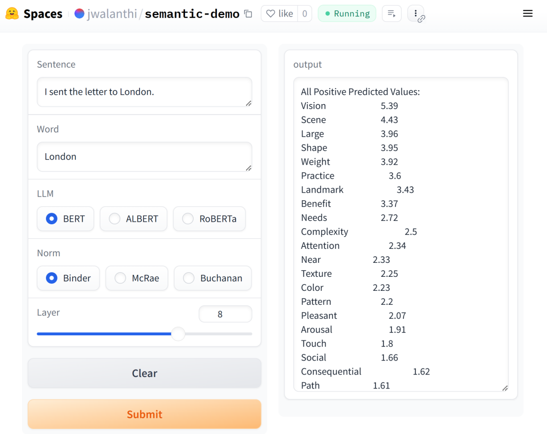
Jwalanthi Ranganathan[1] **Rohan Jha**[1] **Kanishka Misra**[1,2,★] **Kyle Mahowald**[1]
[1]The University of Texas at Austin [2]Toyota Technological Institute at Chicago
`{jwalanthi,rjha,kyle}@utexas.edu` `{kanishka}@ttic.edu`

1 Introduction

The advent of distributional semantic embeddings has enabled major progress in the computational understanding of word meaning by enabling precise statistical explorations of semantic spaces (Erk, 2009; Mikolov et al., 2013; Pennington et al., 2014). More recently, the rise of LMs have made it possible to study embeddings of words in *context*. Chronis et al. (2023) developed a method for projecting *contextual word embeddings (CWEs)* into a interpretable semantic feature space defined by one of three different semantic norms (Binder et al., 2016; Buchanan et al., 2019; McRae et al., 2005). This is achieved by training feed-forward models which map from CWEs from BERT to a vector whose values correspond to feature norms.

Our goal in this paper is twofold: first, we introduce `semantic-features`[1] as an extensible, easy-to-use library based on Chronis et al. (2023) for studying word embeddings from any LM in context. Second, we show its ease of use through an online application which researchers can use without additional programming. We demonstrate these tools with a linguistic experiment that uses this method to measure the contextual effect of the choice of dative construction (prepositional or double object) on the semantic interpretation of utterances.

The dative construction has been of particular interests to theoretical (Goldberg, 1995; Hovav and Levin, 2008; Beavers, 2011) and computational linguists (Bresnan, 2007; Hawkins et al., 2020; Liu and Wulff, 2023; Jumelet et al., 2024; Misra and Kim, 2024; Yao et al., 2025). This is primarily due to its several interesting properties such as its participation in alternation behavior (Levin, 1993), flexible interpretation of the event it describes—caused motion vs. caused possession (Goldberg, 1995; Hovav and Levin, 2008; Beavers, 2011), and interesting feature specific preferences that humans demonstrate while choosing between two dative constructions during production (Bresnan, 2007).



Sentence
I sent the letter to London.

Word
London

LLM
● BERT ○ ALBERT ○ RoBERTa

Norm
● Binder ○ McRae ○ Buchanan

Layer 8

output

```
All Positive Predicted Values:
Vision          5.39
Scene           4.43
Large           3.96
Shape           3.95
Weight          3.92
Practice          3.6
Landmark          3.43
Benefit         3.37
Needs           2.72
Complexity          2.5
Attention         2.34
Near          2.33
Texture         2.25
Color         2.23
Pattern         2.2
Pleasant          2.07
Arousal           1.91
Touch         1.8
Social          1.66
Consequential             1.62
Path          1.61
```

Figure 1: Interactive Demo in Use

Our case study focuses on the semantics of the arguments of the dative construction—in particular its *recipient* argument (Beavers, 2011; Petty et al., 2022). Specifically, we hypothesize that "London" in "I sent **London** the letter." (*double object*; DO) should be more likely to be interpreted as an animate referent (e.g., as the name of a person) than in "I sent the letter to **London**." (*prepositional object*; PO) This is because the DO dative is more canonically associated with possession transfer events, which constrains the recipient to be animate (Beavers, 2011). The PO dative, on the other hand, is associated with both possession transfer and 'caused-motion' (Goldberg, 1995) and allows for inanimate recipients. We test whether LMs learn this distinction by projecting the embedded representation from the token "London" into a more interpretable semantic space and analyze it for animate vs. inanimate features. We include a full demonstration of how to easily obtain such measures from models that have already

[★] Work partly done at UT-Austin before joining TTIC.
[1] `semantic-features` is available at `https://github.com/jwalanthi/semantic-features`

been trained, in addition to describing our full system for training projections from scratch.

2 semantic-features

Our extensible system for training models and analyzing embeddings performs three main tasks: embedding extraction, model training, and hyperparameter tuning. Below, we summarize our methodology; more details can be found in the README.

Embedding Extraction The first step is preparing the CWEs which serve as the 'source' for the training data. Given a (user-provided) corpus and an LM whose weights/embeddings are accessible, `semantic-features` extracts an embedding for each word in the corpus using `minicons` (Misra, 2022). We average the embeddings across all contexts to obtain one vector per word, as in Chronis et al. (2023). While any LM can theoretically serve as the source for word embeddings, autoregressive LMs like GPT-2 are not well-suited for this application because their embeddings only capture left-context for a given word.[2]

Model training All models use a multi-layer perceptron (MLP) to perform feature prediction. All hyperparameters can be user-specified except for the MLP architecture. While Chronis et al. (2023) experimented with other architectures, we choose MLPs to maintain a fully neural system end-to-end. Models are trained with a 80-20 train-validation split, and loss is calculated as mean-squared error between the predicted vector and the ground-truth feature-norm vector.

Hyperparameter tuning Our system allows for hyperparameter tuning by using `optuna` (Akiba et al., 2019). We specifically use the `TPESampler` module, which searches for the combination of hyperparameters which minimizes validation loss using a Tree-Structured Parzen Estimator algorithm. `optuna` searches for the optimal values for hidden size, batch size, and learning rate over a specified set of ranges in Table 3. If enabled, the `MedianPruner` is used to determine which trials to prune. After running 100 trials, the model with the lowest validation loss is saved.

Interactive Demo An interactive demonstration of a selection of models trained using `semantic-features` is available on HuggingFace Spaces as a Gradio app,[3] shown in Figure 1. Users can retrieve a model which maps from the CWE of a user-specified word in context from any layer of BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), or ALBERT (Lan et al., 2020) to any of the three semantic feature spaces used by Chronis et al. (2023).[4] The models were trained using the British National Corpus as the source text. For each word that has a pre-defined feature vector, `semantic-features` extracts the embeddings in each context provided by BNC, averaging the embeddings per word across all contexts. This serves as the source for training, and the feature vector itself serves as the target. Further training details, including hyperparameter specification and GPU training hours, are provided in Appendix B. The output of the demo is a list of the predicted features sorted greatest to least.

3 A Small Case Study on Recipient Semantics in Dative Constructions

Using the tools developed in the previous section, we ask if LMs are sensitive to context-dependent semantics in linguistic constructions. Consider the dative alternation: some ditransitive verbs can take two different argument structures. The first is the *double object (DO)* construction and the second is *prepositional object (PO)* construction.

(1) a. I sent **London** the letter. *DO*
b. I sent the letter to **London**. *PO*

While both are near synonymous, they apply different contextual constraints on their arguments. For instance, in the PO, *London* takes on its "standard" definition as an inanimate place/location, but in the DO, it seems that *London* is an animate recipient (Beavers, 2011; Hovav and Levin, 2008). To what extend do LMs learn this distinction? To test this, we project embeddings from LMs to the Binder features (Binder et al., 2016) space. We choose the Binder Norms here specifically because each feature has a concrete definition provided by the researchers, which can allow for finer grained personhood vs. place-hood distinction. We use these definitions (reproduced in Table 1) to identify Binder features which capture place-hood (Landmark and

[2] As a note, while `semantic-demo` provides the ability to train these MLPs, it does not provide the raw training data itself. Users must obtain corpus data and feature norm data from their respective sources.

[3] Available at `https://huggingface.co/spaces/jwalanthi/semantic-demo`

[4] While these are the models we currently focus on, in principle this can be applied on any masked language model.

Feature	Definition
Biomotion	showing movement like that of a living thing
Body	having human or human-like body parts
Human	having human or human-like intentions, plans, or goals
Face	having a human or human-like face
Speech	someone or something that talks
Landmark	having a fixed location, as on a map
Scene	bringing to mind a particular setting or physical location

Table 1: Feature definitions from Binder et al. (2016).

Feature	DO	PO
Biomotion	**1.19**	0.43
Body	**1.00**	0.26
Human	**0.89**	0.48
Face	**0.71**	0.19
Speech	**0.68**	0.13
Landmark	1.83	**3.43**
Scene	2.59	**4.43**

Table 2: Relevant Binder features predicted for "London" in (1) using CWEs from BERT layer 8. The PO construction lends itself more towards "location" features, and the DO more towards animate features.

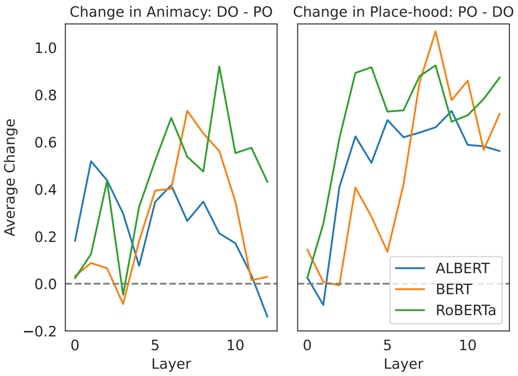


Figure 2: For each layer of an LM, we extract the CWE and project it into Binder space. **Left:** we measure the average change across the test sentences in Person features from PO to DO. The positive values indicate that recipients in the DO are found to be more animate. **Right:** we measure the average change across test sentences in Place features from DO to PO. Here, the positive values indicate that the recipients were found to be more place-like in the PO.

Scene) and person-hood (Biomotion, Body, Human, Face, and Speech) to reflect the two possible salient readings. Higher values for a Binder feature from the projected embedding is taken to mean greater activation of the specific feature. We choose features which capture person-hood and place-hood distinctively, not those which are applicable for both readings. For example, the Vision feature, which is defined as "something that you can easily see," can be activated in both contexts, and is therefore not included in either category. We then extract the embeddings for the recipient word each layer of the LM in each context and project them to the Binder space, observing changes in the relevant features. Table 2 shows an example set of predictions for (1) using BERT layer 8. We see that, consistent with our predictions, "London" is construed as more person-like in the DO and more place-like in the PO.

To test this phenomenon more robustly, we use a method similar to the experiment for studying grammatical roles in Chronis et al. (2023), which requires a balanced dataset of *DO* and *PO* sentences. While the dataset provided by Hawkins et al. (2020) is balanced in terms of the two constructions, it is not well-suited to our needs because the variation in recipient animacy is not focused on the place-like versus animate distinction observed in (1). Instead, we generate 450 alternating pairs in which the recipient is interpreted by a human evaluator to be a person in the DO and a place in the PO. We do this by querying ChatGPT to come up with proper nouns that can be interpreted as places or people, ending up with 15 different such names, all of which were manually checked. This included names of states, as in "Dakota", and names of countries, as in "Jordan", in addition to names of cities, as in "London". We then paired them with 6 different alternating verbs (lemma: send, mail, order, bring, fax, deliver) along with a host of corresponding indirect objects which could also be plausibly received by a place or person. Finally, we choose from five different agents (names), leading to our 450 pairs of sentences, each of the form `[agent]` `[verb]`$_{past}$ `[recipient]` `[theme]` for DO and `[agent]` `[verb]`$_{past}$ `[theme]` `to` `[recipient]` for PO. We project the embeddings of the recipients in context from BERT, RoBERTa, and ALBERT to the Binder feature space and average across construction (DO or PO) and feature set (Person or Place). Fig. 2 shows the average change in feature values for person-hood features vs. place-hood features across the alternants of the dative construction. That is, a value of 0.75 in the animacy panel (left) suggests that the average difference in the activation value of the recipient's animacy features in the DO and PO constructions was 0.75 units on a scale of 0 to 6 (as provided by Binder et al. (2016)) with positive values indicating "more animate in DO than in PO." Similar interpretation (though in the reversed direction) can be made for the right panel, which focuses on place-hood change when switching from DO to PO.

Results As expected, almost all of the models predict an increase in animacy in the DO compared to the PO and an increase in place-hood in the PO compared to the DO (Figure 2) across most layers. There are some exceptions where the change in person-hood/animacy features is in the opposite direction, though these are in the tiny minority (i.e., a total of 3 times out of a total 36 possible model and layer combination). Corroborating with Chronis et al. (2023), we observe particularly high activation-change of the relevant features in layers 6–9 as opposed to the final layer, suggesting possible concentration of semantic sensitivity in those layers. Overall, this suggests that the contextually sensitive distributional semantic embeddings of LMs capture subtle changes in semantic interpretation of different related-constructions.

4 Conclusion

Our hope is that both the complete `semantic-features` library for projecting CWEs into semantic spaces and the online demo will facilitate running linguistically informative experiments using contextual word embeddings.

Acknowledgments

Thanks to the three anonymous reviewers for their feedback. We also thank Katrin Erk and Gabriella Chronis for their comments on an earlier draft and the research project in general. $(\mathrm{KM})^2$ acknowledge funding from NSF Grant 2104995 awarded to Kyle Mahowald.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework.

John Beavers. 2011. An aspectual analysis of ditransitive verbs of caused possession in english. *Journal of semantics*, 28(1):1–54.

Jeffrey R. Binder, Lisa L. Conant, Colin J. Humphries, Leonardo Fernandino, Stephen B. Simons, Mario Aguilar, and Rutvik H. Desai. 2016. Toward a brain-based componential semantic representation. *Cognitive Neuropsychology*, 33(3):130–174.

Joan Bresnan. 2007. Predicting the dative alternation. *Cognitive foundations of interpretation/Royal Netherlands Academy of Science*.

Erin M. Buchanan, K. D. Valentine, and Nicholas P. Maxwell. 2019. English semantic feature production norms: An extended database of 4436 concepts. *Behavior Research Methods*, 51:1849–1863.

Gabriella Chronis, Kyle Mahowald, and Katrin Erk. 2023. A method for studying semantic construal in grammatical constructions with interpretable contextual embedding spaces. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 242–261, Toronto, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Katrin Erk. 2009. Representing words as regions in vector space. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 57–65, Boulder, Colorado. Association for Computational Linguistics.

Adele E Goldberg. 1995. *Constructions: A Construction Grammar Approach to Argument Structure*. University of Chicago Press.

Robert Hawkins, Takateru Yamakoshi, Thomas Griffiths, and Adele Goldberg. 2020. Investigating representations of verb bias in neural language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4653–4663, Online. Association for Computational Linguistics.

Malka Rappaport Hovav and Beth Levin. 2008. The english dative alternation: The case for verb sensitivity. *Journal of linguistics*, 44(1):129–167.

Jaap Jumelet, Willem Zuidema, and Arabella Sinclair. 2024. Do language models exhibit human-like structural priming effects? In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14727–14742, Bangkok, Thailand. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations*.

Beth Levin. 1993. *English verb classes and alternations: A preliminary investigation*. University of Chicago press.

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:1907.11692 [cs]*. ArXiv: 1907.11692.

Zoey Liu and Stefanie Wulff. 2023. The development of dependency length minimization in early child language: A case study of the dative alternation. In *Proceedings of the Seventh International Conference on Dependency Linguistics (Depling, GURT/SyntaxFest 2023)*, pages 1–8, Washington, D.C. Association for Computational Linguistics.

Ken McRae, George S Cree, Mark S Seidenberg, and Chris McNorgan. 2005. Semantic feature production norms for a large set of living and nonliving things. *Behavior research methods*, 37(4):547–559.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv e-prints*, page arXiv:1301.3781.

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

Kanishka Misra and Najoung Kim. 2024. Generating novel experimental hypotheses from language models: A case study on cross-dative generalization. *arXiv preprint arXiv:2408.05086*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP 2014*, pages 1532–1543.

Jackson Petty, Michael Wilson, and Robert Frank. 2022. Do language models learn position-role mappings? In *Proceedings of the 46th annual Boston University Conference on Language Development*.

Qing Yao, Kanishka Misra, Leonie Weissweiler, and Kyle Mahowald. 2025. Both direct and indirect evidence contribute to dative alternation preferences in language models. *arXiv preprint arXiv:2503.20850*.

A Hyperparameters

The following search ranges are used by optuna for hidden size, batch size, and learning rate when enabled.

Hyperparameter	**Lower Limit**	**Upper Limit**
Hidden Size	*min*	min(2**min*, *max*)
Batch size	16	128
Learning Rate	10^{-6}	1

Table 3: Search ranges for optimization, where *min* denotes the minimum between the length of the embedding and length of the feature vector and *max* denotes the maximum between the two values

B Demo Models

All 117 models available through the Gradio app have 2 layers with 50% dropout, and early stopping after 6 epochs of non-decreasing validation loss. The maximum epoch limit was set to 100, though in reality, the best performing models finished training after 40-60 epochs. Hyperparameter tuning was used for hidden size, batch size, and learning rate, and pruning was not enabled. For the Buchanan models, the raw feature labels were not used, and the normalized feature values were used. In total, training all 117 models took 25 GPU hours, including those which were discarded in the process of optimization. Models were trained using an NVIDIA A40 GPU.

Unlocking finite-state morphological transducers: Derivational networks for Inuit-Yupik languages

Coleman Haley
University of Edinburgh
coleman.haley@ed.ac.uk

1 Introduction

While morphology has received substantial attention in computational linguistics and typology, inflectional resources have long out-classed derivational datasets despite growing interest. UniMorph 4.0 (Batsuren et al., 2022), and Universal Derivations (Kyjánek et al., 2020) contain derivational information for 30 and 21 languages respectively, dwarfed by UniMorph's 169 languages for inflection. Further, the typological diversity of languages covered is still limited and dominated by high-resource (Indo-)European languages, with many of the world's most morphologically rich languages (such as so-called polysynthetic languages) entirely excluded from existing datasets.

While existing derivational datasets are limited in terms of typology and language resource status, there is another, closely related resource available for a much broader array of languages: finite-state morphological transducers (FSMTs). These models encode both lexical and morphological information and exist for a wide range of languages, especially very low-resource, morphologically rich languages. This information is stored in a very different form than existing inflectional and derivational morphological resources, however, and is typically not viewed as a dataset, but as a tool.

In this work, we explore the possibility of using FSMTs to create derivational morphology datasets. We focus on the Universal Derivations (UDer) format. This format is richer than that of UniMorph, capturing not just derivationally-related pairs, but the tree structure of entire derivationally-related families of forms. This makes it particularly suitable for capturing derivational information in highly agglutinative, morphologically-rich languages. In this work, we focus on the Inuit-Yupik language family. These languages are known for having an extremely high degree of synthesis, while being heavily agglutinative, and have frequently been cited as canonical examples of polysynthesis, with a higher type-token ratio than any other language family (Park et al., 2021). Further, several languages in the family (kal, ess, iku, esu) have FSMTs publicly available. We produce Universal Derivations-style datasets for Greenlandic (kal; ∼44,000 speakers) and Saint Lawrence Island (SLI) Yupik (ess; ∼500 speakers), using publicly available FSMTs and small text corpora. We make our code and derivational networks in Universal Dependencies format available online.[1]

2 Method

Most FSMTs are primarily designed for morphological analysis; as such, they may generate forms which, while seemingly valid, do not occur (e.g. paradigm gaps). To avoid including such items in our derivational networks, we use existing text corpora for the two languages and use the FSTMs to *analyse* these corpus–thereby restricting us to attested surface forms. We use the digital corpus of SLI Yupik[2], consisting of ∼300,000 unannotated tokens and ∼1,000 manually annotated tokens, and the monolingual Greenlandic corpus collected by Jones (2022), comprising 1.98 million tokens. We use (Chen and Schwartz, 2018)'s FSMT for SLI Yupik and the Apertium morphological analyser for Greenlandic to provide morphological analyses for the corpora (Forcada and Tyers, 2016). For ambiguous words in the SLI Yupik corpus, we use the first analysis from the transducer.

As described in Figure 1, our method works by first analyzing words in the corpus, then repeatedly modifying the analysis and generating forms matching the modified analysis.

Universal-Derivations-style derivational networks typically present words in their standard-

[1] https://github.com/ColemanHaley/fst2dernet/

[2] https://github.com/SaintLawrenceIslandYupik/digital_corpus

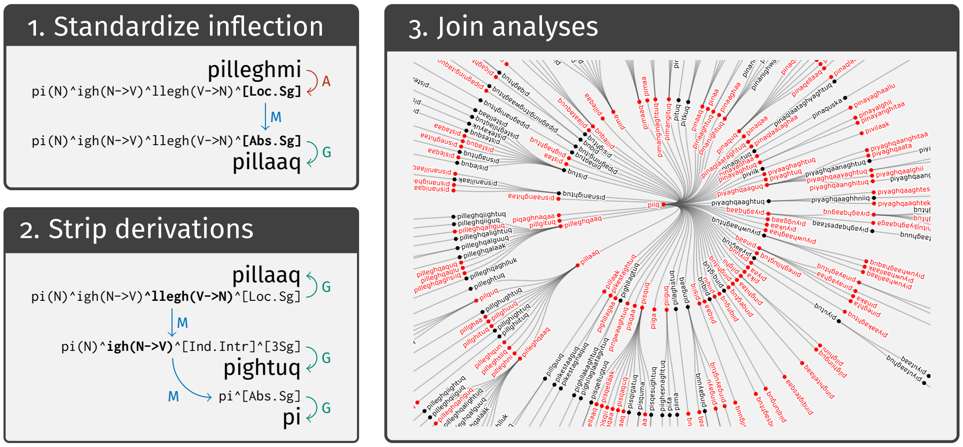

1. Standardize inflection
pilleghmi
pi(N)^igh(N->V)^llegh(V->N)^[Loc.Sg]
A
M
pi(N)^igh(N->V)^llegh(V->N)^[Abs.Sg]
G
pillaaq
2. Strip derivations
pillaaq
pi(N)^igh(N->V)^llegh(V->N)^[Loc.Sg]
G
M
pi(N)^igh(N->V)^[Ind.Intr]^[3Sg]
G
pightuq
M
pi^[Abs.Sg]
G
pi
3. Join analyses

Figure 1: Our method for producing derivational networks from FSMTs. Words in a corpus are first analyzed (A) using the FSMT. We then modify (M) the analysis to have standard inflectional features, and then use the FSMT to generate (G) the standardized form. Next, we recursively modify to strip derivations and generate intermediate forms, producing a chain of derivationally related words. We join chains of derivationally related words to form a network. Red lexemes are attested in the corpus, while black forms are inferred from attested derived forms.

ized or citation forms. In Inuit-Yupik languages, this is the 3rd person singular indicative form of verbs[3] and the absolutive singular form of nouns. For words in our corpora in non-standardized inflections, we feed a standardized version of the analysis to the FSMT to produce the citation form of the word. We treat clitics as special derivational morphemes which occur after inflections.

We now have standardized, analysed forms for all the words in the corpora. But how to go from these to derivational families? We note that an analysis containing several morphs implies the existence of intermediate words, regardless of whether they were seen in the corpus. Because Greenlandic and SLI Yupik are exclusively suffixing, there is no ambiguity about the sequencing of morphs. We can therefore recursively strip off one derivational morpheme at a time to produce a new word. Checking for the part of speech implied by the rightmost derivational morpheme, we add back on the appropriate standardized inflectional features to the analysis, and use the FSMT to produce a surface form for this word if it is unobserved.

We release generalized code for this procedure as well as versions specialized to the analysis format of each of the two FSMTs used here. Our generalized code allows users to specify the formatting of inflectional features, part of speech, clitics, and derivational morphs in the analysis, as well as the set of default features for each part of speech, allowing the extension of our method to other languages with suffixing morphology. Future work could extend our method to languages with both prefixing and suffixing derivation with the use of a model or rule-based system to determine the order of morpheme application/scope.

3 Results

Our derivational networks cover 53,245 lexemes for SLI Yupik and 127,663 lexemes for Greenlandic, on par or surpassing highly-resourced European languages such as Dutch, French, Italian, and English.Further, these lexemes are spread across 6,344 (SLI Yupik) and 11,088 (Greenlandic) distinct derivational families. In contrast to less rich languages, a *majority* of these families are non-trivial (containing at least two lexemes): 4,256 and 6,021; respectively. Further, in both languages almost 1 in 10 derivational families contained 20 or more lexemes (599 `ess`; 1,015 `kal`). The largest derivational families in each language contain many hundreds of lexemes: 359 for the neutral root *piiq* in SLI Yupik, and 1,584 for Greenlandic, far surpassing any single lexeme in existing UDer languages. Finally, we note an impressive range of unique derivational relations/morphemes covered: 397 in SLI Yupik and 327 in Greenlandic.

While this data cannot be considered gold-standard, existing FSMTs and small corpora can

[3] Inuit-Yupik languages mark transitivity inflectionally, productively forming transitive and intransitive variants of verbs. However, because this is a common paradigm gap, we retain the observed transitivity of verbs.

yield large, empirically-grounded derivational networks for extremely low-resource morphologically rich languages. These networks could serve to speed up native speaker annotation, or as silver-standard data in certain types of analysis. These findings corroborate the noted derivational richness of Inuit-Yupik languages. Future work could focus on improving these networks, extending to other languages, building tools for human annotators, or refining these techniques for language with ambiguous morpheme sequencing or parts of speech.

Acknowledgements

I would like to thank Zhifan Sun for his work on creating visualizations of the derivational networks. I would also like to thank Itamar Kastner and the students in the Spring 2023 Morphology course at the University of Edinburgh for feedback on an early version of this project. This work was in part supported by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1) and the University of Edinburgh, School of Informatics and School of Philosophy, Psychology & Language Sciences. Any errors are my own.

References

Khuyagbaatar Batsuren et al. 2022. UniMorph 4.0: Universal Morphology. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 840–855, Marseille, France. European Language Resources Association.

Emily Chen and Lane Schwartz. 2018. A morphological analyzer for St. Lawrence Island / Central Siberian Yupik. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Mikel L. Forcada and Francis M. Tyers. 2016. Apertium: a free/open source platform for machine translation and basic language technology. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation: Projects/Products*, Riga, Latvia. Baltic Journal of Modern Computing.

Alex Jones. 2022. Finetuning a Kalaallisut-English machine translation system using web-crawled data. *Preprint*, arXiv:2206.02230.

Lukáš Kyjánek et al. 2020. Universal Derivations 1.0, A growing collection of harmonised word-formation resources. *Prague Bulletin of Mathematical Linguistics*, 115(1):5–30.

Hyunji Hayley Park, Katherine J. Zhang, Coleman Haley, Kenneth Steimel, Han Liu, and Lane Schwartz. 2021. Morphology matters: A multilingual language modeling analysis. *Transactions of the Association for Computational Linguistics*, 9:261–276.

Aspectual classes as lexically-conditioned predictors of aspectual choice

Laurestine Bradford
McGill University
laurestine.bradford@mail.mcgill.ca

1 Introduction

I investigate a distribution-based characterization of lexical aspectual classes.

The *grammatical aspect* of a verb is morphology which reflects either an internal perspective or an external perspective on the time course of an event. For example progressive aspect *was laughing* in *Mary was laughing when I arrived* takes an internal perspective, while perfective aspect *laughed* in *Mary laughed when I arrived* takes an external perspective. Not all verbs are felicitous in all aspects. For example, verbs denoting static situations usually sound worse in the progressive: **I am knowing French*. It has long been theorized that each verb in a language has an *aspectual class* which captures something about the temporal shape of the described event and thereby explains its compatibilities with different grammatical aspects (for an overview, see Filip, 2020). Indeed, past statistical work explicitly comparing aspectual class labels with the distribution of grammatical aspects has found strong statistical effects (Wulff et al., 2009; Hundt et al., 2020; Bardovi-Harlig, 1998; Andersen and Shirai, 1994), and these are thought to facilitate verb morphology acquisition (Wulff et al., 2009; Shirai and Andersen, 1995).

I propose to flip the script, asking to what extent statistical association with grammatical aspect is an adequate *characterization* of aspectual class. I propose that aspectual class is precisely the lexical information which contributes to aspectual choice. Therefore, it should be detectable by statistically computing each lexical item's contribution to aspectual choice.

This builds on proposals by Brent (1991) and Klavans and Chodorow (1992) to treat the stative-nonstative aspectual class distinction as gradient based on association with the progressive. It is similar in spirit, but orthogonal, to the work of Nerbonne and Van de Cruys (2009) who treat aspectual class as characterized by compatibility with temporal adverbials.

2 Method

I fit a Bayesian mixed-effects logistic regression to a corpus of natural spoken and written text in English (Zeldes, 2017). I did not presuppose any lexical aspectual classes for any verbs. Rather, I fit a model predicting aspectual choice (progressive or perfective), and I included lexical item as a predictor. I then used the fit weights for each lexical item to characterize its lexical aspectual class. A regression model allowed me to include other known predictors of aspectual choice in order to balance out their effects - namely, tense, matrix verb aspect, preceding verb aspect, subject type (singular, plural, mass, or none), object type, perfect morphology, voice, adverbs, subordinating conjunctions, verbal particles, genre, "for"/"in" preposition modifiers, and specific document/author. Mixed-effects regression allowed me to take advantage of frequent lexical items without them overpowering the analysis. A Bayesian model allowed me to obtain estimates for the effects of not just lexical item in general, but each individual lexical item. I also allowed effects varying by lexeme of tense, "for"/"in", and subject/object type, as these are known to affect aspectual class behaviour.

I fit the model using the R package BRMS (Bürkner, 2017, 2018, 2021) with four chains of 7,500 sampling steps. Intercept and linear coefficient priors were normal with standard deviation 2.5 and mean either -2.5 (intercept) or 0 (coefficients). Contact the author for data and code.

3 Results

3.1 Non-lexical predictors

I report results for a subset of predictors.

I replicated some results of Hundt et al. (2020): present tense verbs are more often progressive than

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 373-376.
Eugene, Oregon, July 18-20, 2025

past ($p_d > 0.999, p_{ROPE} < 0.001, 95\%CI = [1.10, 1.88]$) or future ($p_d > 0.999, p_{ROPE} < 0.001, 95\%CI = [1.51, 3.18]$) tense. On the other hand, I found a more consistent result of voice than Hundt et al.: active voice facilitates progressive aspect more than passive ($p_d > 0.999, 95\%CI = [1.69, 2.89]$).

Matching findings of Rautionaho and Hundt (2022), verbs immediately preceded by a progressive verb showed more progressive aspect than those preceded by a perfective ($p_d > 0.999, p_{ROPE} < 0.001, 95\%CI = [0.53, 1.16]$). A temporal adverbial headed by “in” decreased the probability of progressive compared with no temporal phrase ($p_d = 0.982, p_{ROPE} < 0.001, 95\%CI = [-5.76, -0.43]$), but a “for” adverbial was not clearly distinguishable from none ($p_d = 0.760, p_{ROPE} = 0.003, 95\%CI = [-1.39, 1.83]$).

I found significant variation by document (st. dev. $p_{ROPE} < 0.001, 95\%CI = [0.46, 0.84]$) and genre (st. dev. $p_{ROPE} < 0.001, 95\%CI = [0.58, 1.44]$), indicating that style and genre affect aspectual choice, as explored in theoretical (e.g. Smith, 2003; Egetenmeyer, 2021) and corpus (e.g. Mavridou et al., 2015) literature.

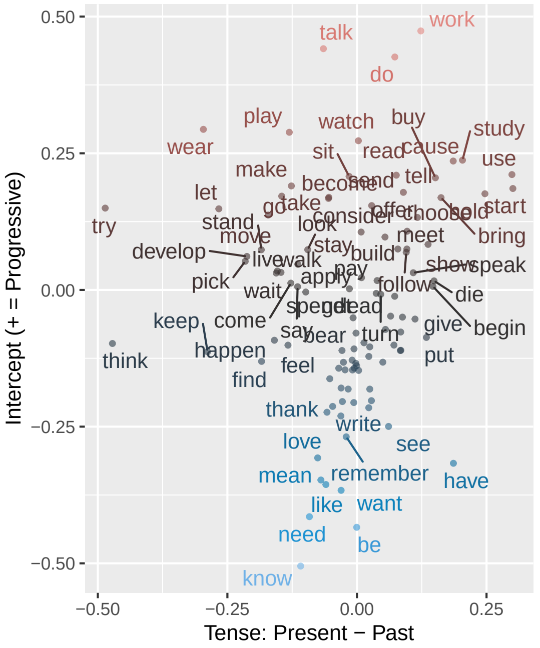


Figure 1: A scatterplot of verbs with at least 25 occurrences. Lexical effect on log-odds of progressive aspect is on the vertical axis and also represented with the color scheme; the horizontal axis shows the lexical effect on the difference in log-odds of progressive between present and past tense contexts.

3.2 Lexical aspectual classes

I found statistically significant variation by lexical item in all measured effects (all st. devs. $p_{ROPE} < 0.001$).

I extracted fit estimates of the effect of each lexical item. All plots in this section are computed for verbs with at least 25 occurrences, and due to computational constraints, they use a subset of 1,000 samples from the model’s posterior distribution.

Figure 1 shows random intercepts representing lexical effect on log-odds of progressive on one axis, and on the other, lexical effect on the present tense vs. past tense contrast. Remarkably, canonically stative verbs exactly coincide with those that strongly disfavor progressive aspect (from *believe*, downward). Moreover, in this plot and others not pictured, stative verbs cluster together in the lexical effects of any other predictors. Thus the lexical aspectual property of dynamicity emerges readily as a predictor of aspectual choice. The verbs which most favor progressive aspect are all standard examples of activities - dynamic verbs with duration but no endpoint.

Meanwhile, the verbs *try*, *think*, and *keep* on the far left of Figure 1, meanwhile, highlight a limitation of my approach. These verbs appear in the past progressive as readily as (or more readily than) in the present progressive. I expect that these verbs are often used to set up background information in a story, a primary use of the progressive aspect (Hopper, 1979). I was not able to control for communicative intent, and it may have contributed to the behaviour of these lexemes.

Not pictured here, lexical items showed a very tight direct relationship between their effect on the present tense vs. past tense contrast and their effect on the future tense vs. past tense contrast. So, the most important axis of lexical variation in tense effect captures how much the past tense specifically favors or disfavors progressive aspect. This is counter to the prediction of a standard theory of aspectual class in which the present perfective (which conceptually forces an event to take place at a single instant) is the most restrictive.

The Bayesian nature of the model allows me to represent uncertainty in its predictions. Figure 2 shows 66% and 95% confidence intervals for lexical effect on log-odds of progressive for a subset of verbs. Due to the small corpus size, the model is not highly confident in any lexical effects.

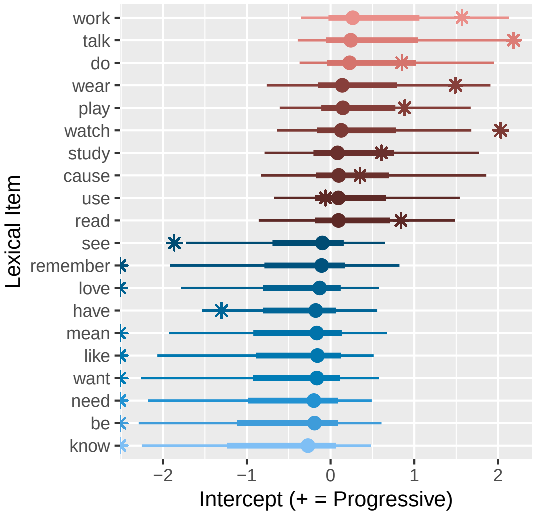


Figure 2: Interval plots showing the ten verbs with highest and lowest fit lexical effect on log-odds of progressive. Intervals show 66% and 95% credible intervals for each lexeme's effect. Star shapes show empirical log-odds-effect of each lexical item computed from raw corpus counts.

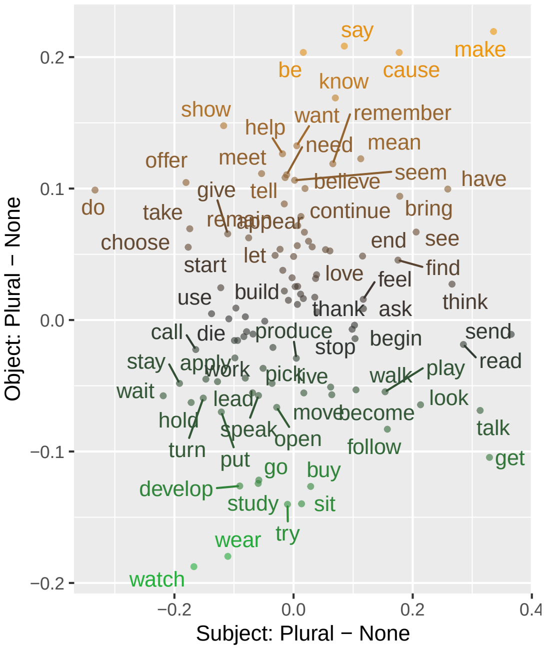


Figure 3: A scatterplot of verbs with more than 25 occurrences. The vertical axis shows lexical effect the difference in log-odds of progressive between contexts with plural objects or with no object, which is also represented with the color scheme; the horizontal axis shows the lexical effect on the difference in log-odds of progressive between plural subjects and no subject.

Figure 2 also shows, with asterisks, the empirical log-odds-ratio of progressive computed for each verb. We see the regularizing effect of modelling. For example, the linear model was able to abstract away from many confounding predictors and identify that *have* lexically behaves like the other stative verbs, despite its differing counts.

Figure 3 shows the lexical effect on the plural object vs. no object contrast on the vertical axis, and the lexical effect on the plural subject vs. no subject contrast on the horizontal. Verbs for which plural objects strongly favor the progressive are mostly of two kinds: stative verbs and verbs of creation and presentation (e.g. *cause* and *bring*). For the former, it is possible that plural objects facilitated an eventive coercion which allowed these verbs to be progressive, possibly by making them gradable. For the latter, this fits with their traditional classification as incremental theme verbs whose aspectual class is linked to their object. Of note, however, is the fact that incremental consumption verbs like *read* do not pattern in the same way.

Not pictured, the words *watch* and "wear" had especially negative plural object vs. singular object contrasts. This suggests that these two words specifically disfavor progressive when they have plural objects. Since each of these usually describes a long sustained interaction with a single object, their use with plural objects may have been restricted to habitual contexts, which disfavor progressive. This, again, is a place where not controlling for communicative intent may have created unexpected results.

For subjects, the lexical effect on plural subject - no subject contrast was closely tied to the lexical effect on the singular subject - no subject contrast (not pictured). This suggests that the largest lexical effect on subject behaviour was in the effect of having no subject. This may have been an oversight on my part: I did not include model the possibility of lexemes varying in the effect of passive voice. Verbs which are on the left in Figure 3 (e.g. *watch*, *do*, *develop*) may just be ones for which the progressive-disfavoring effect of passive voice is less strong.

Finally, the lexical aspectual behaviour of these verbs never appears discretized. We see continuous variation between verbs on all axes. Verbs are known to be able to shift between aspectual classes (Filip, 2020). My data suggest that verbs have different propensities to do this, placing them on a continuum of aspectual behaviour.

4 Outlook

I established that different verbs do contribute differently to aspectual choice, and this effect can be seen in a corpus without incorporating prior knowledge of aspectual classes. This lends support to the existence of aspectual classes (or possibly an aspectual continuum) as well as the potential for children to learn them using their associations with different aspects.

This method could be used to discover aspectual class on a new language. Aspectual class is difficult to discover due to sensitivity to context and brittleness under translation. Our statistical technique does not rely on translation, and so could be used to derive language-internally-motivated aspectual classes. I plan to investigate adaptations to smaller corpora to move toward such an application.

My next steps will be creating a more cognitively-grounded model of aspectual choice. This might follow the model which Gantt et al. (2022) use to derive aspectual categories from survey data or the BayesCat model which Fremann and Lapata (2016) use to learn semantic categories of nouns from a corpus.

References

Roger W. Andersen and Yasuhiro Shirai. 1994. Discourse motivations for some cognitive acquisition principles. *Studies in Second Language Acquisition*, 16(2):133–156.

Kathleen Bardovi-Harlig. 1998. Narrative structure and lexical aspect: Conspiring factors in second language acquisition of tense-aspect morphology. *Studies in Second Language Acquisition*, 20(4):471–508.

Michael R. Brent. 1991. Automatic semantic classification of verbs from their syntactic contexts: An implemented classifier for stativity. In *Fifth Conference of the European Chapter of the Association for Computational Linguistics*, Berlin, Germany. Association for Computational Linguistics.

Paul-Christian Bürkner. 2017. brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, 80(1):1–28.

Paul-Christian Bürkner. 2018. Advanced Bayesian multilevel modeling with the R package brms. *The R Journal*, 10(1):395–411.

Paul-Christian Bürkner. 2021. Bayesian item response modeling in R with brms and Stan. *Journal of Statistical Software*, 100(5):1–54.

Jakob Egetenmeyer. 2021. *Chapter 4 Time Updating Uses of the French Imparfait Extending Across Genres*, pages 56 – 77. Brill, Leiden, The Netherlands.

Hana Filip. 2020. *Lexical Aspect (Aktionsart)*, pages 1–61. John Wiley & Sons, Ltd.

Lea Frermann and Mirella Lapata. 2016. Incremental bayesian category learning from natural language. *Cognitive Science*, 40(6):1333–1381.

William Gantt, Lelia Glass, and Aaron Steven White. 2022. Decomposing and recomposing event structure. *Transactions of the Association for Computational Linguistics*, 10:17–34.

Paul J. Hopper. 1979. *Aspect and Foregrounding in Discourse*, pages 211 – 241. Brill, Leiden, The Netherlands.

Marianne Hundt, Paula Rautionaho, and Carolin Strobl. 2020. Progressive or simple? A corpus-based study of aspect in World Englishes. *Corpora*, 15(1):77–106.

Judith L. Klavans and Martin Chodorow. 1992. Degrees of stativity: The lexical representation of verb aspect. In *COLING 1992 Volume 4: The 14th International Conference on Computational Linguistics*.

Kleio-Isidora Mavridou, Annemarie Friedrich, Melissa Peate Sørensen, Alexis Palmer, and Manfred Pinkal. 2015. Linking discourse modes and situation entity types in a cross-linguistic corpus study. In *Proceedings of the First Workshop on Linking Computational Models of Lexical, Sentential and Discourse-level Semantics*, pages 12–21, Lisbon, Portugal. Association for Computational Linguistics.

J. Nerbonne and T. Van de Cruys. 2009. *Detecting Aspectual Relations Quantitatively*, pages 159 – 182. CSLI Publications. 2009/j.nerbonne/pub003.

Paula Rautionaho and Marianne Hundt. 2022. Primed progressives? predicting aspectual choice in world englishes. *Corpus Linguistics and Linguistic Theory*, 18(3):599–625.

Yasuhiro Shirai and Roger W. Andersen. 1995. The acquisition of tense-aspect morphology: A prototype account. *Language*, 71(4):743–762.

Carlota S. Smith. 2003. *Modes of Discourse: The Local Structure of Texts*. Cambrige University Press.

Stefanie Wulff, Nick C. Ellis, Ute Römer, Kathleen Bardovi-Harlig, and Chelsea J. Leblanc. 2009. The acquisition of tense–aspect: Converging evidence from corpora and telicity ratings. *The Modern Language Journal*, 93(3):354–369.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Empirical Analysis of Russian Aspectual Prefixes: A Computational Approach to Productivity & Semantic Relatedness

Natalia Tyulina
CUNY Graduate Center
ntyulina@gradcenter.cuny.edu

Abstract

This work presents a computational analysis of the productivity of Russian aspectual prefixes. Using multiple complementary methods, including the Tolerance Principle (TP), Baayen's hapax-based measures (P and P*), and semantic similarity scores, we evaluate the extent to which different perfectivizing prefixes are synchronically productive. We construct a large-scale verb lexicon annotated for aspect, and leverage multiple corpora to identify novel prefixed word forms. Our findings reveal that productivity is not uniformly distributed across prefixes: some, like *za-* and *po-*, are frequent and semantically broad, while others, such as *niz-/nis-*, are rare and exhibit narrow unproductive usage, with most appearing productive. Finally, we examine the relationship between productivity and semantic transparency using cosine similarity, finding little evidence that meaning preservation drives rule productivity in the case of Russian prefixes.

1 Introduction

The morphological productivity of aspectual prefixation in Russian has been a subject of longstanding debate. In Slavic languages, the grammatical aspect is encoded in verbal morphology, distinguishing between perfective (PF) and imperfective (IMPF) actions. The IMPF aspect often correlates with atelicity, indicating events that do not have an inherent end-point or culmination, while the perfective aspect indicates completion. Perfectivizing prefixes, those that attach to IMPF base forms to derive PF verbs, are considered the most common morphological process for forming PF verbs in Russian (Forsyth, 1972). However, it remains unclear to what extent these prefixes function as productive processes. While the meanings of some derived verbs can be interpreted compositionally, others exhibit varying degrees of idiosyncracy. Frequency (Bauer, 2001), semantic coherence (Aronoff, 1976) and the ability to produce new forms (Hockett, 1954) are the three criteria for productivity that are often mentioned in the literature. Through a series of computational experiments, we measure the productivity of prefixation in forming PF verbs from simple IMPF verbs. Specifically, we assess:

- The productivity of perfectivization via prefixation.
- The semantic relatedness of PF verbs to their base forms.
- The correlation between productivity measures and semantic relatedness, as well as between productivity measures and a neologisms baseline.

2 Methodology

To quantify morphological productivity, we employ corpus-based and dictionary-based approaches. Specifically, we use two measures based on hapax-legomena, introduced by Baayen (Baayen, 1992): between-rule productivity P* and within-rule P productivity, along with the Tolerance Principle (TP) (Yang, 2005). Additionally, we propose a modified version of P* that incorporates dis-legomena (i.e., terms that occur exactly twice in the corpus), to capture potentially novel low-frequency forms. Furthermore, we compute TP using dictionary-based counts, while Baayen-style metrics rely on corpus statistics. This allows us to compare rule productivity from both a usage-driven and a lexicon-driven perspective.

2.1 Lexicon and Corpus Preparation

We begin by compiling a verb lexicon of 32,489 unique lemmas, each annotated for aspect. This lexicon is based on two sources: an online version of the Russian Morphological Dictionary [1] and a

[1] https://github.com/sshra/database-russian-morphology

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 377-381.
Eugene, Oregon, July 18-20, 2025

precompiled Russian lemma lexicon based on the Grammatical Dictionary of the Russian Language (Zalizniak, 1977). The resulting combined lexicon is used to compute the TP threshold per prefix. Separately, we use a tagged corpus in CoNLL-U format (Nivre et al., 2016) to compute corpus-based derived statistics for Baayen-based measures. Due to computational expenses of processing the full Russian dataset, a subset of approximately 3,500,000 sentences was randomly selected. A variety of genres and styles are represented in the subset, from social media posts to literature passages and technical documents. Subsequently, we parse each sentence using the SynTagRus treebank of Russian model (Nivre et al., 2008) from DeepPavlov (Burtsev et al., 2018) trained on the UD corpora.

Prior to computing productivity measures, we preprocess the data by extracting simple verb forms and the prefixes they occur with, along with their aspects, for both approaches. We repeat the process of prefix extraction twice to account for prefix stacking. We also create a separate category for bi-aspectual verbs that have identical surface forms in PF and IMPF, and treat them as having a null prefix. We only use this category in TP computation for now. Additionally, we remove secondary IMPF verbs from our final set. As a result, we obtain valid prefix-aspect pairs, as defined in both implementations. We also map allomorphs of a given prefix to the same underlying form (e.g., *nis*- and *niz*- or *s*- and *so*-), giving us a total of 23 high-level prefixes.

2.2 Productivity Metrics

As a theory of rule learning, TP establishes a threshold for how many exceptions a productive rule can tolerate. Unlike frequency-based heuristics, TP models the cognitive plausibility of generalizations, explaining, for example, how minority rules, such as certain German plural patterns (Yang, 2016), can still be productive. The TP threshold is given by the formula:

$$\Theta_N = \frac{N}{\ln N}$$

where N is the total number of candidate items (in this case, all simple verb lemmas derived with a given prefix), and e is the number of exceptions (that is, IMPF verbs prefixed with the same prefix). A prefix is considered productive if and only if:

$$e \leq \Theta_N$$

In other words, a prefix that surpasses the threshold in forming PF verbs compared to IMPF verbs is likely productive under TP.

To complement the TP-based analysis, we calculate two metrics derived from the corpus-based statistics. The idea behind this approach is that, since productive affixes tend to give rise to novel words, their frequency distributions are likely to contain a large number of low-frequency forms. Therefore, a good estimate of the affix productivity might be computed as a proportion of low-frequency forms associated with the affix. Then P* is the proportion of all hapaxes in the corpus that are attributed to rule *r*, out of all hapaxes in the corpus. P, on the other hand, is the proportion of all words in the corpus that are attributable to *r* and appear only once, out of all words attributable to *r*. P* is used to compare the differential productivity of various affixes, while P measures the growth rate of the words derived via *r*.

To establish a baseline for productivity, we target neologisms chosen from the online dictionary of Russian neologisms of the 21st century.[2] Most reported neologisms are recent borrowings from English related to social media or technological concepts (e.g., *guglit'* 'to be googling' IMPF; *zaguglit'* 'to have googled' PF). We then compile a separate corpus for each neologism using the Russian web corpora database [3] to retrieve sentences containing the neologisms' base forms preceded by a given prefix. A full list of the neologisms used is provided in Appendix A. We also leverage the CC-100 dataset for Russian (Wenzek et al., 2020) to compute Baayen's productivity statistics and semantic similarity.

For semantic analysis, we use the pretrained neural embedding model DeepPavlov ruBERT (Kuratov and Arkhipov, 2019) to compute contextual embeddings for each token in a sentence, as well as embeddings for each verb from its subtoken components. We compute the average embedding score for each unique lemma and determine cosine similarity scores for each base verb – prefixed verb pair. Finally, we use Spearman's rank correlation to measure the relationship between cosine similarity and productivity metrics, as well as between productivity measures and the neologisms baseline.

[2] https://russkiiyazyk.ru/leksika/slovar-neologizmov.html
[3] https://int.webcorpora.ru/drake/

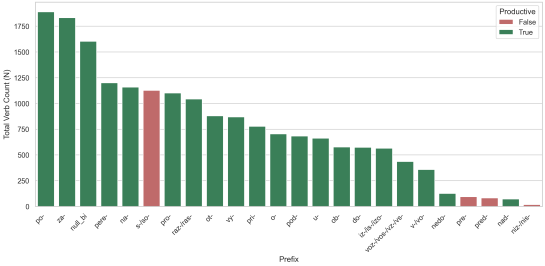


Figure 1: Prefix Productivity Based on the Tolerance Principle

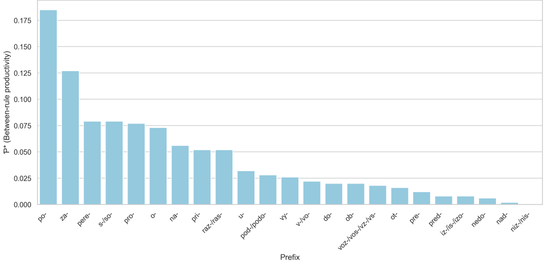


Figure 2: Between-Rule Productivity per Prefix

3 Results

Most prefixes passed the TP threshold, with the exceptions being *s-/so-*, *pre-*, *pred-*, and *niz-/nis-*. Figure 1 shows the TP results by prefix.

In contrast, P and P* yielded different rankings: *pere-*, *pre-*, and *ob-* scored highest under P, while *po-* and *za-* were most productive under P*. Both P and P* assigned *niz-/nis-* a score of zero. Results from between-rule P* measure are presented in Figure 2.

The neologism analysis showed strong alignment with P*, with *po-*, *za-*, and *s-* being dominant in recent coinages. Prefixes such as *niz-/nis-, nad-* and *pred-* were entirely absent from the neologism corpora, reinforcing their low productivity.

Cosine similarity between base and prefixed forms, computed using ruBERT, fell mostly between [0.2–0.37]. Prefixes like *po-* and *nad-* preserved meaning best, while *niz-/nis-* showed the largest shifts. Outlier-rich prefixes like *za-* and *pro-* suggest semantic variability within high-productivity classes. To get a clearer perspective on the distribution of prefixes, we plotted them in Figure 3 using the cosine similarity between each unprefixed–prefixed pair across all verb lemmas.

Finally, visualizing P* against cosine similarity, as shown in Figure 4, revealed no strong linear correlation, underscoring that semantic transparency and productivity do not always go hand in hand. However, both modified P* and P* exhibited the strongest correlation with neologism-based frequency, suggesting their utility in modeling current language trends. Table 1 presents correlation results.

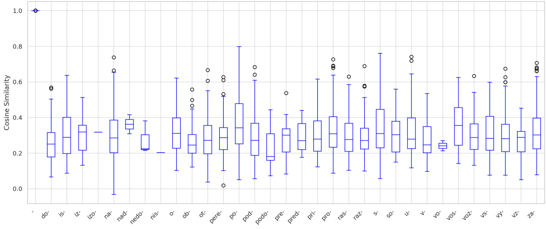


Figure 3: Cosine Similarity per Prefix

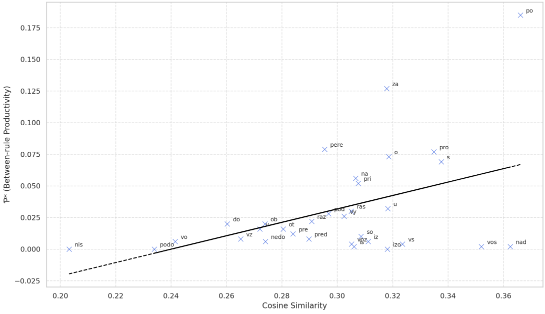


Figure 4: Cosine Similarity vs. Between-Rule Statistics

Correlation Metrics	**Spearman's** ρ
Cosine Similarity & TP	.367
Cosine Similarity & P	$-.065$
Cosine Similarity & P*	.297
Cosine Similarity & Modified P*	.473
Cosine Similarity & Neologisms	.286
Neologisms & TP	.371
Neologisms & P*	.830
Neologisms & Modified P*	.743

Table 1: Spearman correlation coefficients between semantic similarity, productivity metrics, and neologism counts.

4 Discussion

Our results offer empirical evidence that Russian perfectivizing prefixes exist along a productivity continuum. The divergence in rankings across TP, P, and P* illustrates how dictionary-based and corpus-based models capture different nuances of rule behavior. The fact that most prefixes pass the TP threshold underscores their learnability, while hapax-driven P* offers a stronger match to actual coinage trends. TP mainly diverged in its assessment of *s-/so-* and *pre-*, which emerged as unproductive. This warrants further investigation, as finer-grained semantic or phonological features may underlie these patterns and require more careful distinction.

We highlight *niz-/nis-* as a clear outlier across all metrics. It is not seen in any corpus of neologisms and is semantically the most divergent, suggesting that it is unproductive. Meanwhile, *za-* and *po-* illustrate how high productivity can coincide with semantic polysemy. These findings complicate the idea that productivity and semantic transparency necessarily align.

One interesting class of verbs that merits further attention is bi-aspectual verbs—forms that are compatible with both PF and IMPF contexts without overt morphological marking. In our analysis, we incorporated these verbs into the TP framework using a *null* prefix. Surprisingly, this class emerged as productive, supporting the idea that even prefix-less surface forms contribute to learnable morphological generalizations. A natural extension of this work would be to further refine how these verbs are integrated into prefixal paradigms.

We also acknowledge certain limitations, particularly around potential false decompositions. Some verbs exhibit suppletive forms or root alternations that can challenge prefix and suffix identification algorithms. While our pipeline attempts to minimize such cases, they could still influence prefix frequency or similarity metrics in subtle ways.

Importantly, our analysis offers a joint evaluation of productivity and semantic relatedness at scale using modern computational tools. While prior work has often assumed productivity as a binary feature or assessed it through intuition, our study quantitatively profiles prefixal behavior across thousands of verbs and aligns that with neologism usage and semantic drift.

5 Conclusions

Our findings suggest that almost all Russian PF prefixes are in fact productive, with one potential exception being both surface forms of the same underlying prefix niz-/nis-. This work provides a large-scale, computational account of aspectual prefix productivity in Russian. By combining Baayen's corpus-based productivity metrics, Yang's TP, and BERT-based semantic similarity, we show that:

- Prefixes differ in productivity, with P* best predicting real-world lexical innovation.
- Productivity does not always imply semantic transparency; highly productive prefixes like *za-* may exhibit broad or polysemous shifts.
- Discrepancies across metrics point to the need for multiple, complementary perspectives on morphological productivity.

Looking ahead, a deeper investigation into the semantic properties of base verbs—particularly those compatible with *s-/so-* and *pro-*, which were deemed unproductive by TP—may uncover finer-grained subregularities that the current TP-based approach classifies as exceptions. As Yang (Yang, 2023) observes, productivity does not always align with statistical dominance: minority patterns can be highly productive when conditioned by specific features. In our case, such conditioning is likely tied to semantic and phonological factors. Pursuing more granular semantic distinctions within each prefix class may therefore reveal minor but genuinely productive subrules that are obscured in aggregate analyses.

References

Mark Aronoff. 1976. *Word Formation in Generative Grammar*. Number 1 in Linguistic Inquiry Monographs. MIT Press, Cambridge, MA.

Harald Baayen. 1992. *Quantitative aspects of morphological productivity*, pages 109–149. Springer Netherlands, Dordrecht.

Laurie Bauer. 2001. *Morphological Productivity*. Cambridge Studies in Linguistics. Cambridge University Press.

Mikhail Burtsev, Artem Seliverstov, Yuri Kuratov, Dmitry Ermilov, Alexey Gureenkov, Denis Svirchev, Dmitry Kruchinin, Mikhail Shuvalov, Mikhail Aseev, Dmitry Ignatyev, and 1 others. 2018. Deeppavlov: Open-source library for dialogue systems. In *Proceedings of ACL 2018, System Demonstrations*, pages 122–127.

James Forsyth. 1972. The nature and development of the aspectual opposition in the russian verb. *The Slavonic and East European Review*, 50(121):493–506.

Charles F. Hockett. 1954. Two models of grammatical description. *WORD*, 10(3):210–234.

Yurii Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language.

Joakim Nivre, Igor M. Boguslavsky, and Leonid L. Iomdin. 2008. Parsing the SynTagRus treebank of Russian. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 641–648, Manchester, UK. Coling 2008 Organizing Committee.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Charles Yang. 2005. On productivity. *Linguistic Variation Yearbook*, 5:265–302.

Charles Yang. 2016. *The Price of Linguistic Productivity: How Children Learn to Break the Rules of LanguageHow Children Learn to Break the Rules of Language*.

Charles Yang. 2023. A user's defense of the tolerance principle: Reply to enger (2022). *Beiträge zur Geschichte der deutschen Sprache und Literatur*, 145:563–579.

A.A. Zalizniak. 1977. *Grammatical Dictionary of the Russian Language*. Firebird Publications, Incorporated.

A Neologisms

Neologism	Gloss	# Sents
spamit’	to spam	13,239
frendit’	to add as a friend	153
guglit’	to google	74,075
yuzat’	to use	37,430
donatit’	to donate	1,642
trollit’	to troll	2,469
čatit’	to chat	3,529
fotkat’	to take pictures	20,583
kserit’	to take a photo copy	155,328
skanit’	to scan	50,803
skrinit’	to screen	6,137
mejkapit’	to do make-up	32
piarit’	to promote	46,034
startapit’	to start-up	70
kopipastit’	to copy-paste	4,487
follovit’	to follow	2,663
lajkat’	to like	27,007
tegat’	to tag	1,576
šedulit’	to schedule	34
bathertit’	to be talked down to	36
hedlainit’	to make into a headline	11
monitorit’	to monitor	12,625
spoilerit’	to spoil (as in spoiler alert)	5,433
kreativit’	to be creative	7,632
brifit’	to brief	17
loginit’(s’a)	to log in	12,175
čekinit’(s’a)	to check in	2,171
tvitit’	to tweet	1,611
kommitit’	to commit	451
instagrammit’	to post on instagram	261

Visual groundedness as an organizing principle for word class: Evidence from Japanese

Coleman Haley Sharon Goldwater Edoardo Ponti
School of Informatics, University of Edinburgh
{coleman.haley,sgwater,eponti}@ed.ac.uk

1 Introduction

Since classical times, one of the fundamental ideas in linguistic theory is that words are divided into categories with shared syntactic and morphological behaviour. Often called "word classes" or "parts of speech", these classes represent an intersection between linguistic form and semantic function. For example, nouns prototypically refer to objects, and verbs to actions or events.

What is the theoretical status of the relationship between meaning and word class? Within any word class in a given language, exceptions to their semantic properties abound. Nevertheless, there is a great degree of cross-linguistic consistency in the relationship between the meaning of lexical items and their syntactic behaviour–the vast majority of languages clearly handle object words differently from action words. Property words also tend to have special morphosyntactic expression across languages, differing from both nouns and verbs. But for each of these distinctions, there are languages where it is not clearly relevant (Bisang, 2010).

How can a theory explain both these strong universal tendencies and well-established deviations from them? Recent work in computational linguistics has attempted to formalize aspects of the relationship between meaning and form (Rauhut, 2023; Haley et al., 2025). In this work, we focus on Haley et al. (2025)'s notion of (visual) *groundedness*. Groundedness formalizes the notion of how much information a word conveys about an utterance's "meaning" in context–how meaningful vs. grammatical a word is. Haley et al. (2025) showed that visual groundedness shows a clear relationship to the distinction between lexical and functional word classes across 30 languages, demonstrating substantial cross-linguistic consistency–the same classes have similar groundedness across languages. Notably, nouns > adjectives > verbs in terms of visual groundedness, despite all being lexical classes.

If word classes are organized in part by the (visual) groundedness of the meanings they express, then variation in word classes should be associated with differences in groundedness of the expressed meanings. In this study, we focus on Japanese property words, which have the unusual property of constituting two formally very distinct word classes, rather than a single "adjective" class. Building on the insight that one of these classes is more formally "nominal" (*na*-adjectives) and one more "verbal" (*i*-adjectives), we hypothesise that we should see analogous trends in function: one class serving more prototypically nominal functions and one more prototypically verbal. In terms of visual groundedness, this corresponds to higher values for the nominal class.

2 Japanese Adjectives

The two[1] word classes in Japanese typically described as adjectives are *i*-adjectives and *na*-adjectives. These classes are clearly distinguished from each other in Japanese in terms of their syntax and morphology:

(1) *yama-ga takai / takakatta.*
mountain-NOM high / high.PAST
'The mountain is/was tall.' **(*i*-adjective)**

(2) *Taroo-ga sizuka da / sizuka datta*
Taro-NOM quiet COP / quiet COP.PAST
'Taro is/was quiet.' **(*na*-adjective)**

While clearly distinct from nouns and verbs, *i*-adjectives have an analogous inflectional paradigm to verbs (inflecting for aspect and polarity) and can take their syntactic position as in (1), but as shown in (2), *na*-adjectives must be combined with the copula like nouns. Both *i*-adjectives and verbs can modify nouns simply by appearing pre-nominally,

[1] Some linguists identify a third major class, which is identically syntactically distributed to nouns, which we do not concern ourselves with here.

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 382-384.
Eugene, Oregon, July 18-20, 2025

but nouns and *na*-adjectives require a (distinct) attributive marker to modify nouns.

This split is not attributable to phonology or semantics, nor is it a conjugation class. Some stems can belong to both classes. Attempts to describe it under existing semantic hierarchies (Morita, 2010; Oshima et al., 2019) have proven largely unsuccessful.

3 Method

Groundedness is formally defined as the pointwise mutual information between a word/linguistic unit in the context of an utterance, and the meaning of that utterance. We focus on *visual groundedness*–representing meaning with an image. This simplifying assumption makes estimating (visual) groundedness with existing datasets and neural models tractable, and has interesting connections to relevant notions like imageability and perceptual strength. In particular, for an image I and word w_t in an utterance $W = w_1, w_2, w_3 ... w_t ...$, we formalise groundedness as:

$$\text{Groundedness}(w_t) = \log p(w_t \mid I, \mathbf{w}_{<t}) - \log p(w_t \mid \mathbf{w}_{<t}) \quad (3)$$

This allows us to compute groundedness as a *difference in surprisal* between an image captioning model and a (domain-matched) language model. In contrast to typical psycholinguistic norms like concreteness and imageability, groundedness is computed at the (word) *token* level. This implies the same word may be more or less grounded in different contexts.

We use the model released by Haley et al. (2025) as a language model and PaliGemma as the image captioning model. We use the `sudachipy`[2] part of speech tagger to tag words as *i*-adjectives and *na*-adjectives. We focus on the Crossmodal-3600 (XM3600) dataset (Thapliyal et al., 2022), because of its high quality of manual captioning.

As noted by Haley et al. (2025), single groundedness estimates can be noisy, so we filter for only adjective types which occur at least 5 times in our corpus. This is especially important as *na*-adjectives are less frequent than *i*-adjectives in our corpus.

4 Results

Across our corpus of 7185 captions, we find 399 *na*-adjective tokens and 3058 *i*-adjective tokens. These tokens belong to 42 *i*-adjective types and 26 *na*-adjective types. On average, the *na*-adjectives display higher groundedness than *i*-adjectives (3.41 vs. 1.98). Our data has a nested structure, with many tokens of a single word type, and this word type influences groundedness independently of word class (*i*-adjective vs. *na*-adjective). To better estimate the effect of word class itself, we use a linear mixed effects model, with fixed effects of position and word class and a random effect for word type. Under this model, we find a significant effect of word class ($p = 0.029$). Specifically, we find that *na*-adjectivehood increases groundedness by 0.89 ± 0.40 bits.

Two terms are used to compute our visual groundedness measure: surprisal under a language model and surprisal under an image captioning model. Is the association between groundedness and the word class distinction above primarily due to one of these terms? Of particular concern is the first term: perhaps *na*-adjectives are just *a priori* more surprising in the linguistic signal (e.g. expressing lower-frequency concepts). If we find a strong correlation between word class and LM surprisal, it may be that the information provided by the image is dominated by these effects. Fitting the same fixed and random effects as before to instead predict LM surprisal, we do not find a significant effect ($p = 0.133, \beta = 1.17 \pm 0.77$). Similarly, we do not find a significant effect of word class on the captioning surprisal alone ($p = 0.591, \beta = 0.38 \pm 0.61$). So it is only through the interaction between these two factors (groundedness) that an association with word class emerges.

5 Conclusion

Together, our results suggest that *na*-adjectives are used to express more visually grounded meanings than *i*-adjectives in Japanese. In contrast to prior work which failed to find a semantic organizing principle for this distinction (Morita, 2010; Oshima et al., 2019), our work suggests that the formal similarities *i*-adjectives and *na*-adjectives display to verbs and nouns respectively are not arbitrary, but reflect their semantic character.

While still exploratory, our results suggest an exciting role for groundedness in computational linguistics. Together with Haley et al. (2025), these results point to the utility of groundedness not just for explaining cross-linguistic *consistency* in word class organization, but also *variation*. Beyond this, groundedness can also be a useful tool for framing

[2]

and answering questions about the relationship between form and meaning in a particular language, not just cross-linguistically. While groundedness is only somewhat correlated with norms like concreteness or imageability, concreteness allows the asking of related questions where such norms are not available–no relevant concreteness or imageability norms exist for Japanese adjectives. Future work should further validate these results on a larger array of words and datasets, and with new and improved models, and also explore such traditional, human-annotated norms.

Acknowledgements

This work was supported in part by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1) at the University of Edinburgh, School of Informatics and School of Philosophy, Psychology, & Language Science.

References

Walter Bisang. 2010. Word Classes. In *The Oxford Handbook of Linguistic Typology*. Oxford University Press.

Coleman Haley, Sharon Goldwater, and Edoardo Ponti. 2025. A grounded typology of word classes. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10380–10399, Albuquerque, New Mexico. Association for Computational Linguistics.

Chigusa Morita. 2010. The internal structures of adjectives in japanese. *Linguistic Research*, 26:105–117.

David Oshima, Kimi Akita, and Shun Sano. 2019. Gradability, scale structure, and the division of labor between nouns and adjectives: The case of japanese. *Glossa: a journal of general linguistics*, 4(1):41.

Alexander Rauhut. 2023. *Quantitative Aspects of the Word Class Continuum in English*. Ph.D. thesis, Freie Universität Berlin.

Ashish V. Thapliyal, Jordi Pont Tuset, Xi Chen, and Radu Soricut. 2022. Crossmodal-3600: A Massively Multilingual Multimodal Evaluation Dataset. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 715–729, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Self-Supervised Speech Representations in a Pre-train Speech Model Represent Key Rapid Automatized Naming Variability in Autism

Sarah Ethridge[1], Joseph C.Y. Lau[1], Bronya R. Chernyak[2], Rob Voigt[1], Matthew Goldrick[1], Joseph Keshet[2], Molly Losh[1]

[1]Northwestern University, [2]Technion Israel Institute of Technology

1 Background

Individuals with autism experience significant difficulties with pragmatic language, with contributing skills often challenging to measure quantitatively with standard tools. Contributing factors to pragmatic difficulties in autism include differences in speech prosody (e.g., rate, rhythm, intonation; Patel et al., 2020), as well as differences in gaze-speech coordination that contribute to observable differences in social communication (Nayar et al., 2018). Together with differences in the phonetic properties of speech noted in autism, these factors may implicate underlying attentional and physiological differences (e.g., articulatory and visual timing) as mechanistic contributors to clinically appreciable and perceptually "odd" communication styles (e.g., reciprocity, turn-taking) in individuals with autism, their first-degree relatives, and individuals with related genetic conditions (i.e., the *FMR1* premutation; Nayar et al., 2018, 2019, 2021). Thus, fine-grained and accurate characterization of speech in autism is important for informing mechanistically focused intervention strategies grounded in a clearer etiological understanding of pragmatic differences in autism.

2 Objectives:

This study used a novel, deep-learning based measure of phonetic similarity derived from the embedding space of Hidden-unit Bidirectional Encoder Representations from Transformers (HuBERT; Hsu et al., 2021), a state-of-the-art pre-trained speech model using self-supervised learning, to represent speech differences manifested in autistic individuals relative to non-autistic controls. Variability represented through this measure was examined vis-a-vis established acoustic and performance metrics of speech and language profiles (i.e., speech rate, speech rhythm, speech errors, naming time) in autism. The ability for HuBERT to capture further variability in latent, higher order factors of autism, such as modulation of visual attention, was examined using metrics of attentional coordination of speech and gaze.

3 Methods

Analyses included speech samples from 50 autistic individuals and 45 non-autistic controls from the rapid automatized naming (RAN) task, which involved naming serial arrays of common numbers, letters, colors, and objects as quickly and accurately as possible. RAN is a deceptively simple but powerful cognitive measure that indexes speech, gaze, and their integration with important implications for pragmatic language skills in autism. Building on Chernyak et al. (2024) and Kim et al. (2025), error-free, word-sized speech samples from RAN trials were projected into the high-dimensional perceptual space of HuBERT, without the need for pre-selecting acoustic features of interest or manual alignment of speech and text samples. The distance of autistic speech samples from identical non-autistic speech samples was computed using dynamic time warping between embeddings from the 8th transformer layer of HuBERT, based on equivalent model performance across the 8-12th layers in our prior work (Chernyak et al., 2024). Using Pearson's correlations, average distance metrics were analyzed for associations with acoustic (i.e., speech rhythm and rate; Tilsen & Arvaniti, 2013), performance-based (i.e., naming time, speech error rate; Nayar et al., 2018), and gaze metrics of RAN (i.e., visual regressions, perseverations) to examine the potential link between HuBERT distance measures and the attentional coordination of speech and gaze.

4 Methods

Analyses revealed that the HuBERT distance metric was significantly correlated with the following RAN metrics: speech error rate (r (48) = 0.366, $p < 0.01$), speech rate (r (48) = -0.316, $p < 0.05$), naming time (r (48) = 0.531, $p < 0.001$), and visual regressions (r (48) = 0.424, $p < 0.01$; see

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 385-387.
Eugene, Oregon, July 18-20, 2025

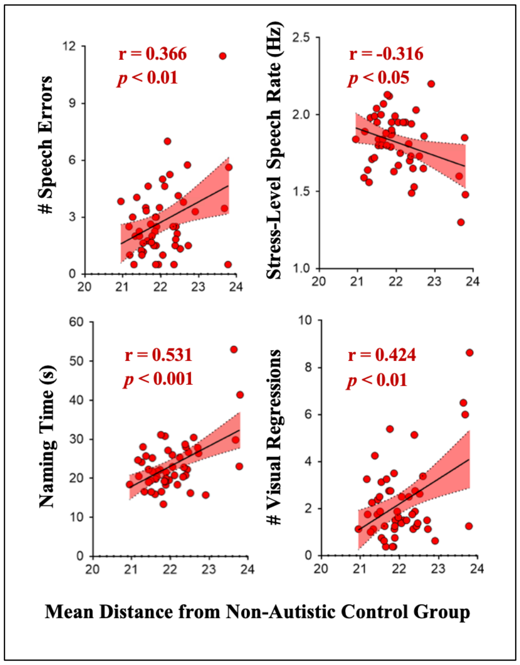


Figure 1: Associations bewteen HuBERT distance and performance, speech, and gaze metrics of rapid automatized naming in autism.

Figure 1). All significant findings survived Bonferroni correction for multiple comparisons. Variability captured by HuBERT speech representations may index subtle prosodic differences in pitch, voice quality and intensity, and articulatory variability subserving higher-order speech and language characteristics of autism, including atypical speech rhythm. Results also suggest that speech representations of HuBERT not only capture meaningful variability of speech in autism but also co-vary with eye gaze patterns that speak to the measure's sensitivity in tapping latent, higher-order linguistic and cognitive factors contributing to the communication profiles of autism.

5 Conclusions

This study demonstrates the potential utility of self-supervised pre-trained speech models, such as HuBERT, which does not require pre-defined acoustic features or speech-to-text alignment, to capture nuanced variability in the linguistic patterns of autism. The results show clear associations with meaningful variability in speech and gaze coordination, underscoring the feasibility of automating linguistic assessments in clinical settings while also providing insights into speech and its multidimensional, cross-modal relationships with broader cognitive processes in autism.

References

Chernyak, B. R., Bradlow, A. R., Keshet, J., & Goldrick, M. (2024). A perceptual similarity space for speech based on self-supervised speech representations. *The Journal of the Acoustical Society of America*, *155*(6), 3915–3929. https://doi.org/10.1121/10.0026358

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., & Mohamed, A. (2021). HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 3451–3460. https://doi.org/10.1109/TASLP.2021.3122291

Kim, S.-E., Chernyak, B. R., Keshet, J., Goldrick, M., & Bradlow, A. R. (2025). Predicting relative intelligibility from inter-talker distances in a perceptual similarity space for speech. *Psychonomic Bulletin & Review*. https://doi.org/10.3758/s13423-025-02652-2

Nayar, K., Gordon, P. C., Martin, G. E., Hogan, A. L., La Valle, C., McKinney, W., Lee, M., Norton, E. S., & Losh, M. (2018). Links between looking and speaking in autism and first-degree relatives: insights into the expression of genetic liability to autism. *Molecular Autism*, *9*, 51. https://doi.org/10.1186/s13229-018-0233-5

Nayar, K., Kang, X., Xing, J., Gordon, P. C., Wong, P. C. M., & Losh, M. (2021). A cross-cultural study showing deficits in gaze-language coordination during rapid automatized naming among individuals with ASD. *Scientific Reports*, *11*(1), 13401. https://doi.org/10.1038/s41598-021-91911-y

Nayar, K., McKinney, W., Hogan, A. L., Martin, G. E., La Valle, C., Sharp, K., Berry-Kravis, E., Norton, E. S., Gordon, P. C., & Losh, M. (2019). Language processing skills linked to FMR1 variation: A study of gaze-language coordination during rapid automatized naming among women with the FMR1 premutation. *PloS One*, *14*(7), e0219924. https://doi.org/10.1371/journal.pone.0219924

Patel, S. P., Nayar, K., Martin, G. E., Franich, K., Crawford, S., Diehl, J. J., & Losh, M. (2020). An Acoustic Characterization of Prosodic Differences in Autism Spectrum Disorder and

First-Degree Relatives. *Journal of Autism and Developmental Disorders*, *50*(8), 3032–3045. https://doi.org/10.1007/s10803-020-04392-9

Tilsen, S., & Arvaniti, A. (2013). Speech rhythm analysis with decomposition of the amplitude envelope: characterizing rhythmic patterns within and across languages. *The Journal of the Acoustical Society of America*, *134*(1), 628–639. https://doi.org/10.1121/1.4807565

Pragmatic Competence in LLMs: The Case of Eliciture

Dingyi Pan and **Andrew Kehler**
Department of Linguistics, UC San Diego
La Jolla, CA, 92093
{dipan|akehler}@ucsd.edu

Large language models (LLMs) consistently produce coherent and meaningful sentences and discourses, hence demonstrating impressive linguistic abilities. Various studies have examined these abilities to assess the extent to which they parallel known properties of human language interpretation. Whereas much of this research has focused on evaluating their syntactic and semantic abilities, fewer studies have examined their skills in the domain of pragmatics. Problems in pragmatics pose unique challenges to LLMs due to their heavy dependence on inference, world knowledge, and context (Chang and Bergen, 2024), and indeed results of previous studies have been mixed. On the one hand, early transformer models like GPT-2 struggle with scalar implicatures and presupposition (Cong, 2022) and fail at detecting and evaluating discourse coherence (Beyer et al., 2021). On the other hand, Hu et al. (2023) found that more recent large-scale language models achieved high accuracy in pragmatic tasks that involve reasoning about the intended meaning of the speaker.

In this paper, we evaluate LLMs on a novel type of pragmatic enrichment that Cohen & Kehler (2021) term CONVERSATIONAL ELICITURE. Consider (1a), which invites the addressee to infer that not only are the children detested by Melissa and are arrogant and rude, but that they are detested by Melissa because they are arrogant and rude.

1. (a) Melissa detests the children who are arrogant and rude. [IC, ExplRC]
   (b) Melissa detests the children who live in La Jolla. [IC, noExplRC]

Note that this inference is not triggered by any syntactic relationship or other type of linguistic felicity requirement that applies to the sentence. Thus, unlike other more commonly studied pragmatic inferences where sentence felicity is at stake (e.g., implicature, presupposition), elicitures are non-mandated. This can be seen in (1b), which is perfectly felicitous despite the fact that it will not typically convey an eliciture that casually relates Melissa's detesting to where the children live.

Previous psycholinguistic studies have demonstrated that people use eliciture inferences in sentence processing tasks such as relative clause (RC) attachment (Rohde et al., 2011; Hoek et al., 2021) and pronoun interpretation (Kehler and Rohde, 2019). Here, we ask two questions regarding the pragmatic abilities of LLMs: Whether LLMs draw elicitures (Exp. 1), and whether LLMs are able to leverage elicitures to guide downstream syntactic processing (Exp. 2).

1 Experiment 1: Detecting Elicitures

Models. We evaluated the performance of eight LLMs: three closed-source models (GPT-3.5-turbo, GPT-4, and GPT-4o) and five open-source models (GPT-2, Llama-3.2-1B, Llama-3.2-3B, and the instruction-tuned versions of the latter two models). The pragmatic abilities of the closed-source models are evaluated via prompting. Since results yielded by prompting might not be an accurate reflection of the underlying linguistic abilities of interest (Hu and Levy, 2023), we evaluate the inferential behavior of the five open-source models by measuring the log probability of a continuation (described below).

Stimuli. We used 60 sets of items in a 2x2 design varying whether the verb in the matrix sentence is an implicit causality (IC) verb (e.g., *detest* in (1)) or non-IC verb (e.g., *babysit* in (2)), and whether the relative clause (RC) conveys a causal eliciture in the IC condition (ExplRC, e.g., *"who are arrogant and rude"* in (1a)) or not (noExplRC, e.g., *"who live in La Jolla"* in (1b)). Since both the IC verb and the explanation RC are required to draw the eliciture inference, the ExplRCs that give rise to an eliciture in the IC variants are not intended to do so in their corresponding non-IC variants (2a).

2. (a) Melissa babysits the children who are arrogant and rude. [nonIC, ExplRC]
   (b) Melissa babysits the children who live in La Jolla. [nonIC, noExplRC]

Tasks. For the closed-source models, we presented each model with the target sentence and explicitly

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 388-391.
Eugene, Oregon, July 18-20, 2025

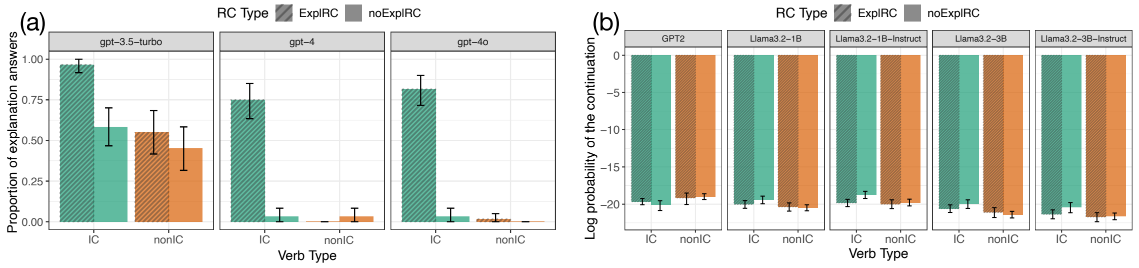


Figure 1: Proportion of explanation answers given by the three closed-source models (a) and the log probabilities of the continuation assigned by the three open-source models (b). The error bars represent 95% confidence intervals.

asked it if the sentence contains an answer to why the event in the matrix clause occurred. We measured the number of "yes" responses to the question and confirmed that the explanation provided by the model matches the content of the RC.

For the open source models, we used the same 240 sentences with the continuation *", and I don't know why."* appended to the end. This continuation should have a lower log probability (i.e., higher surprisal) if the model has inferred that the RC answers the *why*-question via causal eliciture. We summed the log probability of each token in the continuation, including punctuation.

Results. The results are shown in Fig. 1. All closed-source models revealed evidence of inferring elicitures. Although GPT-3.5-turbo overgenerated elicitures, it was still more likely to infer that sentences with IC verbs and paired explanation RCs provided answers to the *why*-question. GPT-4 and GPT-4o show a much stronger pattern, whereby they almost exclusively produced explanation answers in the IC/ExplRC condition.

Turning to the open-source models, results from the Llama models revealed that in IC contexts, the continuation was less likely in sentences in which the RC provides an explanation than when it does not. Further, there is a reliable effect of verb type in the noExplRC conditions, suggesting that when the RC does not provide an explanation, the continuation is less likely for nonIC verbs than for IC verbs. This result is expected since non-IC verbs are less likely to prompt an expectation for an explanation of the event in the matrix clause (Kehler et al., 2008) and hence raise the question *Why?*. Thus, explicitly stating "I don't know why" is predicted to be more surprising in the nonIC/noExplRC condition than in the IC/noExplRC condition. Lastly, the interaction between verb type and RC type was significant, suggesting that the type of RC affected IC verbs more than nonIC verbs. In contrast, GPT-2 showed none of the predicted effects. In sum, these results suggest that all Llama models were able to draw the eliciture inference, but not GPT-2.

Discussion. All closed-source models provided more explanation responses in the IC/ExplRC condition than in the other three conditions. Further, all Llama models showed the effects of verb and RC content as well as their interaction on the continuation that expresses the ignorance of the cause, suggesting that regardless of the model size and instruction-tuning, these models are able to draw eliciture inferences. In contrast, GPT-2 does not show any patterns that would suggest the inference of eliciture. This result is in line with previous findings of a large improvement in performance on pragmatic tasks for models with greater than 1B parameters (Hu et al., 2023).

One might worry that the expected patterns we observed in the model performance are not due to the inference of eliciture, but are instead driven by the establishment of lower-level (e.g., word) associations. We believe this interpretation is unlikely given our 2x2 design. Specifically, since sentences in the IC/ExplRC and IC/noExplRC conditions minimally differ in the content of the RC, the observed differences in the model responses and log probabilities cannot be attributed solely to the properties of IC verbs. Similarly, sentences in the IC/ExplRC and nonIC/ExplRC conditions have the same RC but different verb types, and thus the differences between conditions cannot be solely driven by the RC either. Taken together, the results suggest that all closed-source models and the Llama models show the ability to draw elicitures.

Since all of the models besides GPT-2 show evidence of being able to draw elicitures, our findings raise the question of whether these models can leverage them to guide syntactic processing. In Exp. 2, we examine the effect of eliciture in a case study using ambiguous RC attachment.

2 Experiment 2: Anticipating Elicitures

Background. Rohde et al. (2011) reported on an experiment using examples like those in Exp. 1, except where the direct object of the main verb is a complex NP containing singular and plural NPs as possible attachment sites for an ensuing RC (3).

3. (a) Melissa babysits the children of the musician who is/are ...
   (b) Melissa detests the children of the musician who is/are ...

The well-documented low-attachment bias in English predicts that the auxiliary *is* in (3a), which agrees in number with the lower NP, will be read faster than *are*, which agrees with the higher NP (Frazier, 1978; Carreiras and Clifton, 1999, *inter alia*). However, Rohde et al. (2011) predicted that this bias would shift toward high attachment for (3b), due to (i) IC verbs creating a high expectation that an explanation will ensue, (ii) that an ensuing RC might provide one through eliciture, and (iii) any such explanation would be about the direct object of the matrix verb, which is the high attachment option for the RC. Their predictions were confirmed. Here we examine whether LLMs show evidence of the same behavior.

Models. The behaviors of the same models examined in Exp. 1 were evaluated.

Stimuli. We modified the 60 stimulus sets from Exp. 1 to take the form of (3). We counterbalanced and randomized the order of the two noun phrases, such that half of the items have the plural NP as the high attachment site, and half have the singular NP as the high attachment site.

Tasks. For the closed-source models, we presented each of the two auxiliaries as possible continuations and asked the model to select between them. The order of the answer choices, reflecting either the high or low attachment bias, was balanced across items.

For the open-source models, we obtained the raw probability of each auxiliary and calculated the log-odds ratio by taking the difference, i.e., $\log(p_{high}) - \log(p_{low})$. Higher log-odds ratios indicate a greater model bias toward high attachment.

Results. The results are shown in Fig. 2. There was a significant effect of verb type for GPT-4, showing a greater high attachment preference with IC verbs than with nonIC verbs. Neither GPT-3.5-turbo nor GPT-4o showed the expected high attachment preference for IC verbs. For the open-source models, the log-odds ratio obtained from all Llama models was higher for IC sentences than nonIC ones, suggesting that the high attachment preference is stronger with IC contexts. GPT-2 did not show a difference in attachment preference between the two verb types.

Discussion. Among the three closed-source models, only GPT-4 shows an increase in the high-attachment preference when an IC verb is used than when a non-IC verb is used. Even though GPT-3.5-turbo and GPT-4o exhibited evidence of drawing elicitures when explicitly prompted in Exp. 1, neither showed a significant difference in the attachment preference between the two verb types. A possible reason for this finding is that GPT-4's performance is enabled by having more parameters than the other two models. This hypothesis remains speculative, however, since the number of parameters and the specifications of the model architectures have not been made public. In addition, the non-significant results might be a by-product of the prompting task, since prompting may require additional metalinguistic knowledge, and hence model performance may not always align with raw probabilities that reflect linguistic abilities (Hu and Levy, 2023).

On the other hand, among the five open-source models, all Llama models showed a stronger bias for the high attachment site when an IC verb is used than when a nonIC verb is used. Together with the results in Exp. 1, this suggests that these models can not only infer elicitures but also anticipate them as a source of information when processing the RC. In contrast, GPT-2 does not show the expected pattern, suggesting that it lacks the ability to use pragmatic inferences in RC attachment decisions. This result is likely due to its failure to draw elicitures in the first place, as demonstrated in Exp. 1.

3 General Discussion

The pattern we observe shows that larger and more recent LLMs demonstrate the greatest sensitivity to the presence of eliciture. On the one hand, the negative results for GPT-2 cast doubt on its ability to draw elicitures, aligning with prior studies showing at-chance performance on other pragmatic tasks (Beyer et al., 2021; Hu et al., 2023). At the same time, our findings contribute to the positive evidence of the pragmatic abilities of more recent LLMs. In Exp. 1, the three closed-source models were all able to detect eliciture in the IC/ExplRC

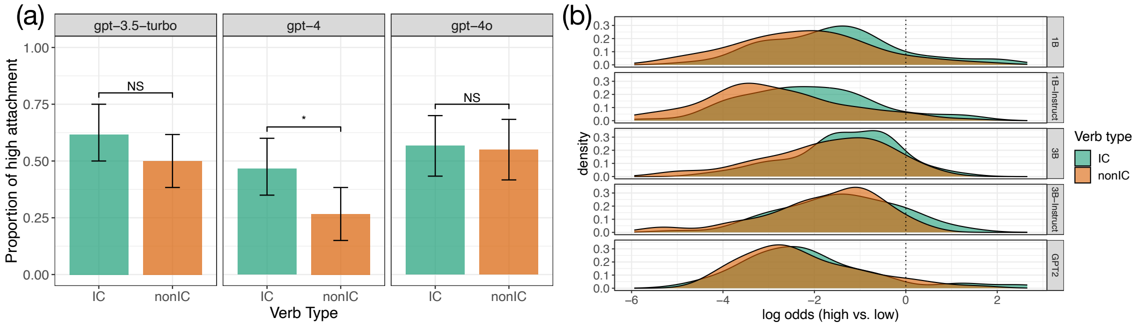


Figure 2: The proportion of responses that show high attachment bias in the three closed-source models (a) and the log-odds ratio between the probability of the critical word that reflects either high or low attachment bias in five open-source models (b). The error bars represent 95% confidence intervals.

condition, although GPT-3.5-turbo overgenerated elicitures to varying extents in the other three conditions. Similarly, all four Llama models revealed the predicted interaction whereby the log probabilities of the continuation *", and I don't know why."* were lower in the IC/ExplRC condition than the others.

In terms of the use of pragmatic inference in syntactic processing, the results of Exp. 2 suggest that the Llama models were also able to make predictions about ensuing elicitures, which in turn enabled them to make predictions about a syntactic attachment decision, as reflected by the relevant preference for a specific word (i.e., auxiliary). Moreover, even though all closed-source models were able to draw the eliciture inference when prompted, only GPT-4 displayed evidence that the anticipation of eliciture impacted the prediction of an auxiliary in the IC condition, reflecting a greater bias toward high attachment compared to the non-IC condition. Further research with other models and larger data sets will be necessary to pin down the properties of LLMs and their training that most contribute to their ability to detect and utilize eliciture.

Acknowledgments

We thank Alex Warstadt for helpful feedback and for suggesting the sluiced prompt used in Experiment 1, and to participants at the Seventh California Meeting on Psycholinguistics for useful discussions.

References

Anne Beyer, Sharid Loáiciga, and David Schlangen. 2021. Is incoherence surprising? Targeted evaluation of coherence prediction from language models. In *Proceedings of NAACL 2021*, pages 4164–4173.

Manuel Carreiras and Charles Clifton, Jr. 1999. Another word on parsing relative clauses: Eyetracking evidence from Spanish and English. *Memory and Cognition*, 27:826–833.

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

Jonathan Cohen and Andrew Kehler. 2021. Conversational eliciture. *Philosophers' Imprint*, 21(12):1–26.

Yan Cong. 2022. Psycholinguistic diagnosis of language models' commonsense reasoning. In *Proceedings of the first workshop on commonsense representation and reasoning (CSRR 2022)*, pages 17–22.

Lyn Frazier. 1978. *On comprehending sentences: Syntactic parsing strategies*. Ph.D. thesis, University of Conneticut.

Jet Hoek, Hannah Rohde, Jacqueline Evers-Vermeul, and Ted JM Sanders. 2021. Expectations from relative clauses: Real-time coherence updates in discourse processing. *Cognition*, 210:104581.

Jennifer Hu, Sammy Floyd, Olessia Jouravlev, Evelina Fedorenko, and Edward Gibson. 2023. A fine-grained comparison of pragmatic language understanding in humans and language models. In *Proceedings of ACL-2023*, pages 4194–4213.

Jennifer Hu and Roger Levy. 2023. Prompting is not a substitute for probability measurements in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore. Association for Computational Linguistics.

Andrew Kehler, Laura Kertz, Hannah Rohde, and Jeffrey L Elman. 2008. Coherence and coreference revisited. *Journal of semantics*, 25(1):1–44.

Andrew Kehler and Hannah Rohde. 2019. Prominence and coherence in a Bayesian theory of pronoun interpretation. *Journal of Pragmatics*, 154:63–78.

Hannah Rohde, Roger Levy, and Andrew Kehler. 2011. Anticipating explanations in relative clause processing. *Cognition*, 118(3):339–358.

CNNs that robustly compute vowel harmony do not explicitly represent phonological tiers

Jane Li and Alan Zhou
Department of Cognitive Science,
Johns Hopkins University
Baltimore, MD
{sli213, azhou23}@jhu.edu

1 Introduction

Linguistic and model-theoretic analyses of long-distance phonology postulate the existence of phonological tiers (Goldsmith, 1976; Heinz et al., 2011). For example, vowel harmony may be analyzed as a process that projects vowels (but not consonants) onto a tier and ensures that all sounds on the tier have the same feature (e.g., [±front] in Turkish vowel harmony, Clements et al. (1982)).

Li and Zhou (under review) recently demonstrated that convolutional neural networks (CNNs) learning a toy example of vowel harmony (§2) on short strings robustly generalize the pattern to much longer strings. One explanation is that these CNNs have independently recovered an "algorithm" that is consistent with the tier projection analysis. Alternatively, these models may have uncovered an approximation of this system, or an entirely different system that robustly generalizes to long lengths. This work investigates these hypotheses via various interpretability methods. In particular, we search for evidence for a "strong" implementation of tier projection, in which these CNNs exactly implement the tier-projection and feature-matching analyses described above.

2 Model and toy language

We follow the architecture of the CNN string recognizer described in [4]. Strings are passed as a block of one-hot character encodings into a convolutional neural network consisting of 4 one-dimensional layers. The output of this CNN is passed through a global max-pool, followed by a single fully connected layer that outputs for each string a binary classification score between 0 and 1. Strings with score above 0.5 are treated as belonging to the recognizer's string language (e.g. the set of strings obeying an unbounded vowel harmony rule). Each convolutional layer is parameterized with a kernel size of 3, a channel size of 256, and a stride of 1 with same padding.

CNNs were trained on an artificial string acceptance task designed to emulate a pattern of transparent unbounded vowel harmony. Artificial strings are sampled by generating syllables roughly obeying the sonority sequencing principle with a vowel inventory {a, e, o, u, ä, ë, ö, ü}, with the constraint of vowels agreeing in the presence of trema (V̈) or absence of trema (V) in harmonious strings. Models learned to recognize if a given string obeys the vowel harmony rule, obtaining perfect test accuracy even over strings much longer than those seen during training.

3 CNNs do not implement exact tier projection

We first investigate the hypothesis that these CNN models are explicitly performing "hard" tier projection. That is, there exists some intermediate layer of the CNN in which vowels (but not consonants) are being projected. If this is the case, we hypothesize that unprojected consonant strings such as [spl] and [spr] should not be distinguishable from one another in terms of activation at that layer. We tested this prediction by decoding the consonants [l] from [r] and the voiceless stops [p,t,k] from each other. For each set of sounds, we selected all attested length-3 consonant clusters where sounds in the set can appear interchangeably. We obtained activations for all of these clusters and decoded the presence of one target sound in the sound set (e.g., [spr] has [p], but [str] and [skr] do not). We find that all sound sets are reliably decodable (Fig. 1A).

Although the performance of the decoder drops off towards later layers, it remains substantially higher than that of a conservative baseline. We observe a similar trend when we attempt to decode sound presence in CVCVC sequences (e.g., is [p] present in [palar] vs. [torel]?). However, we note that while decoding accuracy falls off in later layers for all sounds, consonants consistently fall off

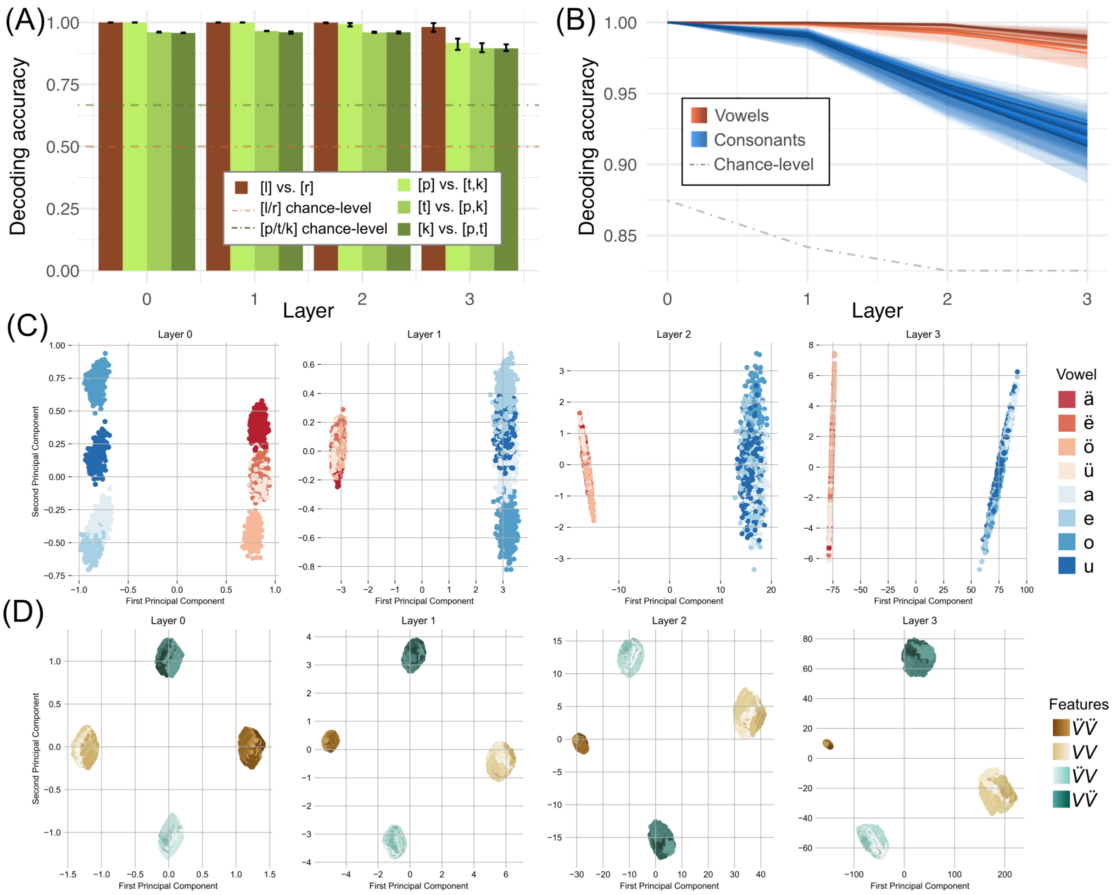


Figure 1: **A.** Decoding accuracy (via ridge regression) for [l] against [r] and [p, t, k] against each other. **B.** Decoding accuracy for presence of individual sounds. Error bars/ribbons in (A, B) indicate range of accuracies across 5 runs. **C and D.** Intermediate activations projected along the first two principal components obtained from PCA over all possible CVC inputs (C) and all possible CVCVC inputs (D). Activations are taken after the application of ReLU in each layer, and flattened for PCA. Each individual set of activations is colored by the identity of the vowel in the CVC sequence (C) and by the sequence of vowel features in the CVCVC sequence (D).

more than vowels (Fig. 1B). While we conclude that these CNNs have not learned to perform tier projection exactly, some prioritization for vowels over consonants is observed.

4 CNNs demonstrate feature-based abstraction over vowels

We now turn to ask whether among the vowels, some abstraction has formed to facilitate the computation of vowel harmony, such as that of V̈ as a category vs. V. We investigate this by applying principle component analysis (PCA) to the activations of each convolutional layer in response to all possible CVC sequences (Fig. 1C), and separately for all CVCVC sequences (Fig. 1D). Applying PCA to the CVC inputs, we find evidence that CNN representations do reflect abstract vowel features, with the V-V̈ distinction being strongly captured by the first principal component (PC) in all layers of the network. Applying PCA to the CVCVC outputs yields similar findings, with the first PC capturing the distinction between the two harmonious feature combinations (VV vs. V̈V̈) and the second PC capturing the distinction between the two disharmonious feature combinations (VV̈ vs. V̈V). We do note, however, that neither of these dimensions seem to reflect the distinction between harmonious and disharmonious feature sequences itself. Preliminary examination suggests that this distinction may be found in the third principal component, though perhaps in a less robust manner than the distinctions described above.

5 Discussion

5.1 A soft implementation of tiers

Altogether the results indicate that the trained CNNs are not implementing an algorithm that fully resembles strict tier projection. However, results do point toward a soft implementation of tiers. Under this hypothesis, the concept of tiers still maps onto a layer of the network, but the layer still has capacity (and learns) to represent other contrasts that are irrelevant to the pattern at-hand. In the case of this toy example, we observe vowel representations become progressively abstract across layers (Fig. 1C) and track vowel bigram information (Fig. 1D), but consonants, which are theoretically irrelevant, are still reliably decodable across all layers (Fig. 1A and 1B). The main prediction is that vowels have "privileged" representations (e.g., better signal within-model) over consonantsthat support computations for the task at hand. This is most evident in the decoding results of Fig. 1B, where vowels consistently better decoded than consonants.

5.2 Alternative theories and their implementations

So far, the possible implementations that have been discussed in this work pertain to a specific framework (tier-based analyses of harmony patterns). It could be the case that the CNNs examined in this study are implementing an algorithm that is consistent with other theories of harmony. Some theories, which assume different forms of input (e.g., articulatory accounts of harmony, Gafos (1999)), may render the models incompatible or be considered as an implementation of intermediary representations. That aside, the methods utilized in this work can be generalized to test hypotheses about what theories a model has learned to implement. A phonological theory makes predictions about what instances (e.g., phonological strings) have shared or contrastive representations. Translating these predictions to signals from model read outs, it predicts that contrastive representations to be decodable or occupy a representational subspace.

5.3 Disambiguating between representations of grammaticality and tier-based representations

We found via PCA that the model has learned to linearly represent the distinction between harmonic and disharmonic vowel sequences. Considering that this is theoretically the only contrast that the model needs to learn to distinguish, these findings are currently confounded with a grammaticality (in other words, output True/False oriented) representation and an algorithmic abstraction of vowel sequences. This should become distinguishable when a model is equipped to learn multiple patterns. Eventually, all patterns have to converge to some representation that supports final True/False decisions, but should have different specific, detectable, representational content for each pattern learned.

6 Conclusion

Our results suggest that these CNNs have converged to a robust solution for unbounded vowel harmony, albeit one that is different from the mechanism of explicit tier projection. In particular, we find that vowels and consonants are both highly decodable from intermediate activations, contrary to what is predicted by an exact tier projection account. However, the intermediate activations of the CNN do reflect robust representations of the vowel features over which harmony is computed, with preliminary evidence for representation of the distinction between harmony and disharmony.

Acknowledgments

We thank the reviewers of this abstract for their helpful feedback. This work is supported by a SSHRC Doctoral Fellowship to JL (752-2024-0291).

References

George N Clements, Engin Sezer, et al. 1982. Vowel and consonant disharmony in turkish. *The structure of phonological representations*, 2:213–255.

Adamantios I Gafos. 1999. *The articulatory basis of locality in phonology*. Ph.D. thesis, Johns Hopkins University.

John Anton Goldsmith. 1976. *Autosegmental phonology*. Ph.D. thesis, Massachusetts Institute of Technology.

Jeffrey Heinz, Chetan Rawal, and Herbert G Tanner. 2011. Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human language technologies*, pages 58–64.

Investigating the Probability of External Causation in Hindi Light Verb Constructions

Kanishka Jain, Ashwini Vaidya
Indian Institute of Technology, Delhi
{kanishka, avaidya}@hss.iitd.ac.in

1 Introduction

The predominant approach to analyze 'causal-noncausal' alternation in linguistics is by showing that one of the forms tends to be more coded (morphologically or phonologically) than the other (Haspelmath, 1993). For instance, in Hindi, the causal (or causative) form of the verb freeze takes the causative morpheme /-va/ (/jəm/ 'freeze' – /jəm-va/ 'caused to freeze'). Hence, the causal form is more coded than the noncausal one in this case.

Haspelmath in his series of works (Haspelmath, 2008; Haspelmath et al., 2014; Haspelmath, 2016, 2021), further extends the idea by introducing the notion of 'form-frequency' correspondence and predictability. He proposes that the form and frequency of a lexical item are correlated such that items that are more frequent are less coded or shorter compared to infrequent items. In case of the causal-noncausal alternation, if the noncausal verb is more frequent than its causal counterpart then the causal form is more coded resulting in a 'causal alternation'. But if it is the causal form that is more frequent, then the noncausal form takes an extra coding which is known as an 'anticausal alternation'. Table 1 shows examples from Swahili.

			gloss	C	NC
causal alternation	*gandisha*	***ganda***	freeze	20	**82**
anticausal alternation	***vunja***	*vunjika*	break	**883**	336

Table 1: C= causal occurrence, NC= noncausal occurrence. Verb pairs from Swahili such that in a causal alternation the noncausal form is more frequent than the causal form, and vice-a-versa in case of anticausal alternation (Haspelmath, 2008).

Causal-noncausal alternations, as in the above table, also reflect on the lexical properties of a verb such that verb pairs forming a causal alternation like 'freeze' are spontaneous events. They occur automatically without any external agent while anticausal alternations like 'break' are non-spontaneous events and occur due to the intervention of an external agent (Haspelmath et al., 2014). For instance, in English when the noncausal verb 'die' changes to the causal verb 'kill' there is an addition of external argument as shown in (1). Here, (1-a) denotes a change of state for the argument 'Sam' but (1-b) expresses the cause meaning such that John caused Sam to die. Hence, valency change is a crucial property of causal-noncausal alternations.

(1) a. Sam died.
b. John killed Sam.

However, the scope of previous studies has been limited to lexical and morphological causative alternations, and the use of other predicates as causatives have been neglected. This work aims at analyzing Light Verb Constructions (LVCs) in Hindi, where nominals alternating with the light verbs /kərna/ 'do' and /hona/ 'be' signal causal and noncausal meaning, respectively (Ahmed and Butt, 2011; Vaidya et al., 2019). Examples are shown in (2) and (3). In (2) noun /cori/ 'theft' appears with the noncausal verb /hui/ 'be' and does not require an external agent. On the other hand when the same noun appears with causal verb /ki/ 'do' as in (3) it takes an external agent /ləDka/ 'boy'. This alternation of meaning and structure is similar to our previous examples in (1).

(2) gehnõ-ki **cori** **hui**
jewellery-GEN.F theft.F be.PERF.F
'There was theft of jewellery.'

(3) lə rke-ne gehnõ-ki **cori**
boy.3.SG.M-ERG jewellery-GEN.F theft.F
ki
be.PERF.F
'The boy stole the jewellery.'

Since, light verb causal alternations as in (2) and (3) are derived from the same lexical item, that is the noun here, Haspelmath (1993) recognize them as 'equipollent' alternations or constructions with 'symmetric' coding that is both forms are coded (Haspelmath, 2021). This is in contrast with other previously investigated phenomena where one form is more coded than the other.

Further, unlike lexical and morphological causatives where the core meaning of an event comes from the verb, in case of LVCs the predicating noun carries the meaning of an event. Hence, properties like type of arguments and their semantic roles (like agent and patient) are also intricately tied to nouns instead of verbs. For instance, the noun /cori/ 'theft' in (1) and (2) is an agentive noun such that even when there is no agent in (1), there is still presupposition that there was an agent of the stealing event. In contrast, Hindi also has nouns like /izafa/ 'increase' that generally do not presuppose an external agent.

(4) ĩd^{h}ən-ki qimat mẽ izafa
fuel.M-GEN.F price.F in increase.M
hua hɛ
be.PERF.SG.M be.PRS.SG
'There is an increase in the price of fuel.'

In Hindi, the argument structure of LVCs is also dependent on the lexical properties of the nouns. For

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 395-397.
Eugene, Oregon, July 18-20, 2025

instance, nouns like /bɛt^{h}ək/ ‘meeting’ in (5) when occurs with the causal verb /kərna/ ‘do’ they take only one argument /məntri/ ‘minister’. While nouns like /vɪcar/ ‘thought’ in (6), when they combine with the same light verb it takes two arguments, /məntri/ ‘minister’ and /prəstav/ ‘proposal’.

(5) kəl məntrɪyõ-ne
yesterday ministers.3.PL.M-ERG
bɛt^{h}ək **ki**
meeting.3.SG.F do.PERF.F
‘The ministers held a meeting yesterday.’

(6) kəl məntrɪyõ-ne
yesterday ministers.3.PL.M-ERG
prəstav-pər **vɪcar** **kɪya**
proposal3.SG.M-on thought.3.SG.M do.PERF.M
‘The ministers considered the proposal yesterday.’

Nouns also have selection restrictions on the light verbs such that not all light verbs can combine with a noun to form an LVC (Butt, 2010). For example, nouns like /yad/ ‘memory’ can occur with different light verbs forming different LVCs (/yad kərna/ ‘intentionally remembering something/someone’, /yad hona/ ‘having a memory of someone/something’, /yad ana/ ‘unintentionally remembering something/someone’) but nouns like /pəres^{h}ani/ ‘trouble’ can only combine with light verb /hona/ (pəres^{h}ani hui ‘had a trouble’).

Considering how nouns affect both the structure and meaning of an LVC, it is interesting to ask if nouns in such constructions also affect the causalness of an LVC in Hindi. Therefore, this paper extends the notion of causality to the predicating nouns. In particular, we ask if the frequently expressed meanings can help us identify a causal or anticausal alternation for the nouns in a light verb construction. This is crucial for identifying the argument structure of the predicating noun and predicting the likelihood of the light verbs it may occur with. This also helps to build lexical resources like subcategorization frames.

2 Encoding Causalness

In this work we are interested in the general likelihood that a noun occurs more with the causal verb /kərna/ ‘do’ or with the noncausal verb /hona/ ‘be’. We show that nouns occurring more frequently with the light verb /kərna/ carry agent-oriented semantics while those that occur more frequently with /hona/ do not.

Similar to previous works (Haspelmath, 2008; Samardžić and Merlo, 2012, 2018), we study the /kərna/-/hona/ alternation by extracting their frequency distribution from syntactically annotated corpus of Hindi. We use the corpus to generate a list of alternating nouns. Following Haspelmath et al. (2014), we then calculate their degree of causalness for an LVC by dividing the total number of /kərna/ alternation multiplied by 100 by the sum of its /kərna/ and /hona/ alternations. Nouns that have high degree of causalness tends to occur more as causatives and nouns with low degree tends to occur more as inchoative. As discussed above, in /kərna/-/hona/ alternation there is no derived or marked form therefore evaluation in terms of form-frequency correspondence is not possible. Hence, to check for the validity and variability of our findings we test for agency and reproducibility. In Hindi, one way to express agency is via using ergative marker /-ne/ on the subject. We test if the agentive nouns have high probability of occurring with the ergative marker than others. We then show that this pattern is observable in other corpus of the language as well. For this, we find a list of commonly occurring LVCs across these corpora under study and found that the predicates have similar distribution.

3 Method and Results

Noun	**gloss**	**Caus**	**Caus**	**%E**	**%E**
		HUTB	**HTB**	**HUTB**	**HTB**
g^{h}oʃŋa	announcement	97.7	83.3	82.9	62.5
fɛsla	decision	93.9	60	74.7	40
palən	compliance	87.5	90	37.5	0.0
ʃadi	marriage	57.1	61.1	42.8	55.6
bɛt^{h}ək	meeting	37.5	66.7	15.6	41.7
prarəmb^{h}	start	25	33.3	25	16.7
izafa	increase	16.7	28.6	8.3	0.0

Table 2: A sample of alternating LVC pairs from HUTB corpus and HTB. Caus=Causalness, %E= percentage of ergatives

To find the LVCs having /kərna/-/hona/ alternation, we have selected the Hindi-Urdu Dependency Treebank (HUTB) (∼ 4m tokens) (Bhatt et al., 2009). HUTB is a manually annotated corpus that already identifies LVCs by using the label ‘pof’ (part-of)’ and therefore LVcs can be automatically retrieved. Since, this work depends heavily on the number of LVCs that we find in the corpus we have taken only the news section ∼ 3.7m tokens) as the size of conversation data (∼ 25k tokens) is too small. We find the frequency of all the LVCs in which the nominal alternates with both the light verbs. A total of 121 alternating LVCs were found. However, to remove any chance occurrence from our analysis we remove pairs with frequency less than 1 for both the alternations giving us a list of 53 LVCs. A sample is shown in Table 2.

Based on their degree of causalness we can see that the nouns at the high end have higher probability of taking an external argument than those at the lower end. This further testified by the percentage of ergatives they occur with.

To check the validity of the realization of causalness for Hindi LVCs we try to find out whether an LVC shows a consistent behavior across different corpora or not. We conducted a comparative study by finding commonly occurring alternations in a different corpus. We compare our previous list of 53 nouns with the Hindi TimeBank’s (HTB) fictional crime part (∼ 0.2m tokens) (Goel et al., 2020). We found 25 such pairs that were common to both the corpora. We can see that nouns do show a general tendency to occur either as a causal item or as an noncausal item across the different corpora (as shown in Table 2).

In order to verify the extent to which ergativity is related to the causalness we’ve also calculated Spearman’s rank correlation coefficient. The coefficient amounts to 0.606 (level of significance = .01 (one-sided)), indicating a robust correlation between the two. However, the correlation coefficient for HTB amounts to 0.323 (level of significance = .01 (one-sided)). There are two reason for low correlation in HTB. First, ergative marker /ne/ in Hindi appears only with the subject of past perfective sentences and as a result this test didn't cover all the instances of the subject. Second, the size of the HTB corpus is smaller in comparison to HUTB.

4 Discussion

In this paper we've investigated nouns alternating with the light verbs /kərna/-/hona/ in terms of their causal property. Constructions like LVCs are distinctive as both the forms are coded therefore Haspelmath's original proposal of form-frequency correspondence and predictability of the shortness of the form does not translate to them[1]. Therefore, in this work we have extended the idea to investigate the property of 'causality' in nouns. We hypothesize that nouns have a preference towards the predicting verb which can be shown using the form-frequency correspondence. Nouns that carry more agent-oriented semantics prefer the causal verb /kərna/ while those that don't prefer the noncausal verb /hona/.

We conduct a corpus study and show that nouns in an LVC indeed have likelihood towards either the causal-noncausal formation. Nouns with high degree causalness encode agent-oriented semantics and tend to occur frequently with causal verb /kərna/ while those with lower values occur with /hona/. This is further verified by the correlation between causalness and ergativity for HUTB. We also found that similar patterns can be attested for the commonly occurring LVCs in a different corpus for Hindi.

However, there were limitations to our work. Since, Hindi has no fixed list for LVCs one may find an instance of an LVC in one corpus but not in others. Second, apart from ergativity, agency can also be tested using other parameters like animacy and volitionality of the subject. Our ongoing work focuses on testing the subject of an LVC on these various parameters. Lastly, unlike previous studies the numbers shown here are from one language only and in future work, we aim to conduct a cross-linguistic study.

References

Tafseer Ahmed and Miriam Butt. 2011. Discovering semantic classes for urdu nv complex predicates. In *Proceedings of IWCS 2011*.

Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, and Fei Xia. 2009. A multi-representational and multi-layered treebank for Hindi/Urdu. In *Proceedings of LAW III*, pages 186–189.

Miriam Butt. 2010. The light verb jungle: Still hacking away. *Complex predicates in cross-linguistic perspective*, pages 48–78.

Pranav Goel, Suhan Prabhu, Alok Debnath, Priyank Modi, and Manish Shrivastava. 2020. Hindi TimeBank: An ISO-TimeML annotated reference corpus. In *16th Joint ACL - ISO Workshop on Interoperable Semantic Annotation PROCEEDINGS*, pages 13–21.

Martin Haspelmath. 1993. More on the typology of inchoative/causative verb alternations. *Causatives and transitivity*, 23:87–121.

Martin Haspelmath. 2008. Frequency vs. iconicity in explaining grammatical asymmetries. *Cognitive Linguistics*, 19(1).

Martin Haspelmath. 2016. Universals of causative and anticausative verb formation and the spontaneity scale. *Lingua Posnaniensis*, 58(2):33–63.

Martin Haspelmath. 2021. Explaining grammatical coding asymmetries: Form–frequency correspondences and predictability. *Journal of Linguistics*, 57(3):605–633.

Martin Haspelmath, Andreea S. Calude, Michel Spagnol, Heiko Narrog, and Elif Bamyacı. 2014. Coding causal–noncausal verb alternations: A form–frequency correspondence explanation. *Journal of Linguistics*, 50:587 – 625.

Tanja Samardžić and Paola Merlo. 2012. The meaning of lexical causatives in cross-linguistic variation. *Linguistic Issues in Language Technology*, 7.

Tanja Samardžić and Paola Merlo. 2018. The probability of external causation: An empirical account of crosslinguistic variation in lexical causatives. *Linguistics*, 56(5):895–938.

Ashwini Vaidya, Owen Rambow, and Martha Palmer. 2019. Syntactic composition and selectional preferences in hindi light verb constructions. *Linguistic Issues in Language Technology*, 17.

[1] A reviewer asked about the efficiency of coding and communication which LVCs seem to violate. According to Haspelmath (2021), constructions like Hindi LVCs are examples of a 'uniformly explicit' coding system where efficiency is less important than the explicit coding of meaning.

The Unnatural Language ToolKit (ULTK):
a software library for research in computational semantic typology

Nathaniel Imel
New York University
n.imel@nyu.edu

Christopher Haberland
University of Washington
haberc@uw.edu

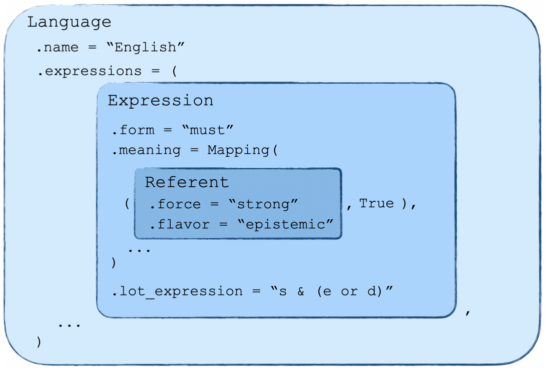
Shane Steinert-Threlkeld
University of Washington
shanest@uw.edu

1 Introduction

This paper introduces the **U**nnatural **L**anguage **T**ool**k**it (ULTK), an open-source Python library for computational semantic typology research (https://clmbr.shane.st/ultk/). ULTK's key features include unifying data structures, algorithms for generating artificial languages, and data analysis tools for related computational experiments. The language module organizes the basic data structures for constructing meaning spaces, expressions, and languages. A grammar submodule contains methods for building and enumerating expressions from custom Language of Thought (Fodor, 1975, 2008; Quilty-Dunn et al., 2022) grammars, which allows for straightforward computation of minimum length descriptions for symbolically expressible semantic representations. This approach has been used successfully in many investigations of concept learning (Feldman, 2000; Goodman et al., 2015). The second main module of ULTK, effcomm, organizes efficient communication analyses, which have become popular styles of explanation in recent functionalist accounts of semantic universals (Kemp et al., 2018). This module contains functions for defining informativity based on literal and pragmatic communicative agents and algorithms for exploring the space of artificial languages.

After first elaborating on the structure of these two modules, we then provide two case studies, illustrating two major styles of explanation in computational semantic typology research: (1) an efficient communication analysis of modal semantic typology, and (2) an analysis of the relative ease of learning of monotone versus non-monotone quantifiers. ULTK's accessible design, documentation, and open-source nature are intended to reduce barriers for researchers when implementing computational linguistic typological experiments.

2 Language module



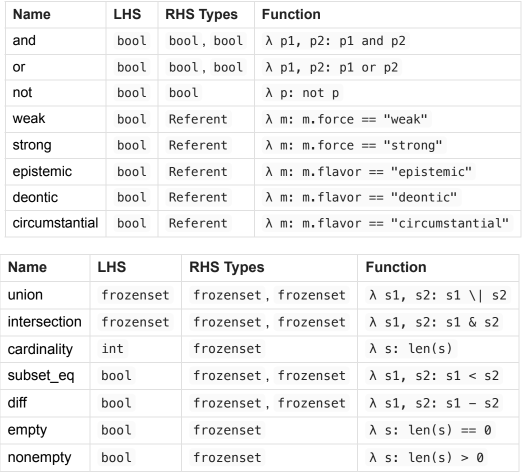
```
Language
  .name = “English”
  .expressions = (
        Expression
        .form = “must”
        .meaning = Mapping(
            ( Referent
              .force = “strong”
              .flavor = “epistemic” , True ),
            ...
        )
        .lot_expression = “s & (e or d)”
                                              ,
     ...
  )
```

Figure 1: Structure of a ultk.language.Language, using the English modal vocabulary as an example.

In ULTK, a Language (Figure 1) is a collection of Expressions; an Expression is a mapping between a surface form and a Meaning; a Meaning maps a Universe's Referents to an object of arbitrary type (e.g., bool if the meaning is boolean). A Referent is a wrapper for any hashable Python object, which could be as simple as an index or as complex as a model-theoretic structure. A Universe is a collection of Referents. In this way, a bool Meaning corresponds to the characteristic function of a set. To capture probabilistic meanings, it is natural to use float meanings.

Grammar This submodule contains classes and functions for building Grammars and generating expressions. These are often used for semantic representations: at its core, this module enables composing functions to arbitrary depth according to their input and output types. A Grammar is made up of arbitrary Rules, with GrammaticalExpressions formed by combining rules with licensed input and output types (Piantadosi, 2014). A Rule minimally consists of a name, a left-hand side (output type) a right-hand side (sequence of input types), a function to apply, and optionally a weight (for defining

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 398-401.
Eugene, Oregon, July 18-20, 2025



Name	LHS	RHS Types	Function
and	bool	bool , bool	λ p1, p2: p1 and p2
or	bool	bool , bool	λ p1, p2: p1 or p2
not	bool	bool	λ p: not p
weak	bool	Referent	λ m: m.force == "weak"
strong	bool	Referent	λ m: m.force == "strong"
epistemic	bool	Referent	λ m: m.flavor == "epistemic"
deontic	bool	Referent	λ m: m.flavor == "deontic"
circumstantial	bool	Referent	λ m: m.flavor == "circumstantial"

Name	LHS	RHS Types	Function
union	frozenset	frozenset , frozenset	λ s1, s2: s1 | s2
intersection	frozenset	frozenset , frozenset	λ s1, s2: s1 & s2
cardinality	int	frozenset	λ s: len(s)
subset_eq	bool	frozenset , frozenset	λ s1, s2: s1 < s2
diff	bool	frozenset , frozenset	λ s1, s2: s1 – s2
empty	bool	frozenset	λ s: len(s) == 0
nonempty	bool	frozenset	λ s: len(s) > 0

Figure 2: The ULTK LoT grammars in our case studies, modals (top) and quantifiers (bottom, snippet).

probabilistic grammars). A Grammar can be initialized by loading a Python module with arbitrary functions parsed as Rules, or by loading a YAML file (Figure 2). The grammar submodule can be used to generate minimum length descriptions for Meanings in order to quantify their representational complexity for computational experiments; this can be done by depth-bounded enumeration (with user-specified uniqueness criteria) or by approximate Bayesian inference over PCFGs.

3 Effcomm module

The effcomm module provides tools for analyzing the communicative efficiency of languages. The agent and informativity submodules implement Rational Speech Act-style agents and enable the computation of literal and pragmatic informativity of languages (Frank and Goodman, 2012; Degen, 2023). These tools for measuring informative communication, together with tools from language.grammar for measuring the complexity of languages, can be combined to study how languages balance, or trade off, various pressures efficiently. The effcomm module also includes submodules for generating hypothetical languages through various sampling strategies (sampling), approximating Pareto-optimal solutions to efficiency trade-offs via an evolutionary optimization algorithm (optimization), and evaluating the languages' communicative properties (tradeoff). The analysis submodule provides utilities for visualizing language distributions in trade-off space. These components are designed to work together to support end-to-end efficient communication analyses of artificial or natural languages.

4 Case study 1: efficient communication for modals

Efficient communication has been proposed as an explanation for variation in semantic typology (Kemp et al., 2018). Using ULTK, we replicate Imel et al. (2024) by applying this analysis to modals. To do this, we (1) convert attested modal vocabularies to ULTK Languages, (2) generate artificial vocabularies, and (3) measure efficiency and a notion of *naturalness*. For the latter , we consider the degree to which a language satisfies the Independence of Force and Flavor (IFF) semantic universal (Steinert-Threlkeld et al., 2023). We define a modal Universe of (force, flavor) Referents and construct Languages as sets of Expressions mapping these referents to truth values (Fig. 1).

Languages Natural vocabularies are derived from a public database (Guo et al., 2022), while artificial ones are generated via ULTK's language.sampling and effcomm.optimization modules. The former samples meanings randomly while controlling IFF satisfaction, and the latter uses an evolutionary algorithm to approximate the Pareto frontier for the complexity/communicative cost trade-off. This step uses some convenience methods that ULTK provides for turning data in fieldwork-natural formats into its natural internal data structures that are needed for an efficient communication analysis; this helps lower the barrier-of-entry to conducting such analyses.

Efficient communication Complexity is measured as minimum description length in a boolean Language-of-Thought (LoT) (Kemp and Regier, 2012), using language.grammar to enumerate and cache shortest expressions. Communicative cost is measured in effcomm.informativity, which models literal communication and uses communicative need priors estimated from English news data. While the results we present here use literal speakers and listeners, ULTK offers convenience methods for iterating pragmatic agents to arbitrary depth from a given language (Frank and Goodman, 2012; Degen, 2023).

Results Figure 3 plots complexity vs. communicative cost, with artificial languages colored by

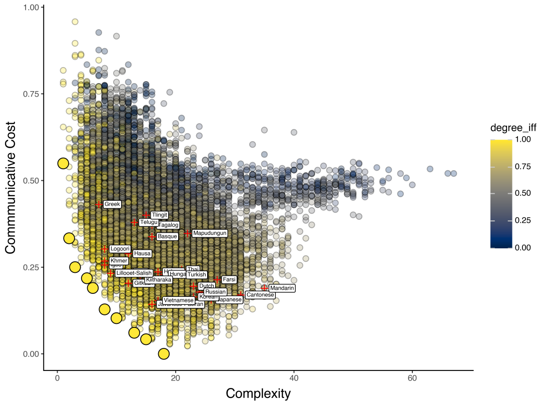


Figure 3: Replication of Imel et al. (2024) via ULTK; see text for details. Full demo available at `https://github.com/CLMBRs/ultk/tree/main/src/examples/modals`.

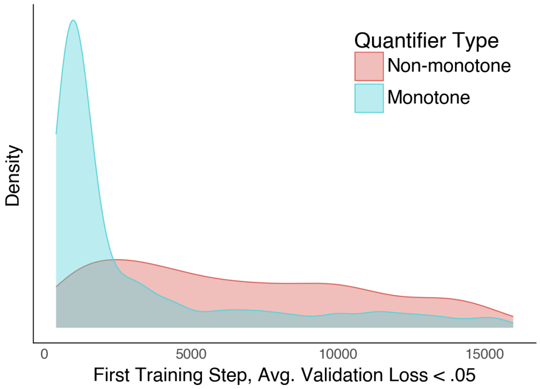


Figure 4: Replication of Haberland and Steinert-Threlkeld (2025) via ULTK; see text for details. Full demo available at `https://github.com/CLMBRs/ultk/tree/main/src/examples/learn_quant`.

naturalness and natural ones marked in red. Natural languages cluster closer to the Pareto frontier (large points) than chance ($t(15536) = 46$, $p \approx 0$), and naturalness negatively correlates with Pareto distance ($\rho = -0.38$, $p \approx 0$). Similar replications of efficient communication analyses (e.g., kinship (Kemp and Regier, 2012), indefinite pronouns (Denić et al., 2022), quantifiers (Steinert-Threlkeld, 2021), connectives (Uegaki, 2021)) are under development. ULTK provides abstractions and utilities that allow for relatively simple replication of these existing analyses and, therefore, makes it easy to conduct new ones as well.

5 Case study 2: ease of learning for (monotone) quantifiers

Semantic *universals* constrain natural linguistic meanings (Croft, 2003). For example, all simple determiners in natural languages are argued to be monotonic (Barwise and Cooper, 1981). A possible explanation is that monotone quantifiers are *easier to learn* (Steinert-Threlkeld and Szymanik, 2019; Chemla et al., 2019). Using ULTK, we replicate one of the results contained in Haberland and Steinert-Threlkeld (2025) (which contains full experimental details), showing that monotone quantifiers are easier to learn than non-monotone quantifiers. We generate a large number of quantifier expressions composed from a LoT grammar (see Figure 2). We measure ease of learning as the number of steps required by a neural model to learn to correctly judge the truth-value of a quantifier. We generate 2000 quantifiers from the LoT grammar and measure the speed at which both LSTM and Transformer models learn to verify expressions that are both monotonic and non-monotonic. We find that monotone quantifiers are typically learned much faster than non-monotone ones (Figure 4). This suggests that ease of learning may be a factor shaping the lexical semantic typology of the world's languages, at least in this domain (see (Steinert-Threlkeld, 2020; Steinert-Threlkeld and Szymanik, 2020; Maldonado et al., 2022; Maldonado and Culbertson, 2019; Strohmaier and Wimmer, 2022) for other case studies in other domains).

This case study demonstrates the potential of using ULTK to answer questions about the relation between semantic universals and ease of learning. In addition to the LoT grammar, the library provides basic tools for structuring `Meaning`s and other objects in a way that is consumable by external machine learning libraries.

6 Conclusion

The **U**nnatural **L**anguage **T**ool**k**it (ULTK) is an open-source library, enabling linguists to execute computational typological research. Our intention is to lower the barrier-to-entry to conducting efficient communication and ease-of-learning analyses of typological phenomena. In the limit, typologists and fieldworkers will be able to input data structured in natural ways, and the library will facilitate analyses in these these domains. The two case studies presented here demonstrate the possibility of this division of labor and the utility of the ULTK library. Future work will expand both coverage of methods and improve the ease of use to continue making this dream a reality. We welcome submissions of contributions, questions, and suggestions to our code repository (`https://github.com/CLMBRs/ultk`).

References

Jon Barwise and Robin Cooper. 1981. Generalized quantifiers and natural language. In *Philosophy, language, and artificial intelligence: Resources for processing natural language*, pages 241–301. Springer.

Emmanuel Chemla, Brian Buccola, and Isabelle Dautriche. 2019. Connecting Content and Logical Words. *Journal of Semantics*, 36(3):531–547.

William Croft. 2003. *Typology and universals*. Cambridge University Press.

Judith Degen. 2023. The Rational Speech Act Framework. *Annual Review of Linguistics*, 9(1):519–540.

Milica Denić, Shane Steinert-Threlkeld, and Jakub Szymanik. 2022. Indefinite Pronouns Optimize the Simplicity/Informativeness Trade-Off. *Cognitive Science*, 46(5):e13142.

Jacob Feldman. 2000. Minimization of Boolean complexity in human concept learning. *Nature*, 407(6804):630–633. Publisher: Nature Publishing Group.

Jerry A Fodor. 1975. *The language of thought*, volume 5. Harvard university press.

Jerry A Fodor. 2008. *LOT 2: The Language of Thought Revisited*, 1 edition. Oxford University Press, Oxford.

Michael C. Frank and Noah D. Goodman. 2012. Predicting Pragmatic Reasoning in Language Games. *Science*, 336(6084):998–998.

Noah D. Goodman, Joshua B. Tenenbaum, and Tobias Gerstenberg. 2015. Concepts in a Probabilistic Language of Thought. In *The Conceptual Mind: New Directions in the Study of Concepts*. The MIT Press. _eprint: https://direct.mit.edu/book/chapter-pdf/2271061/9780262326865_cav.pdf.

Qingxia Guo, Nathaniel Imel, and Shane Steinert-Threlkeld. 2022. A Database for Modal Semantic Typology. In *Proceedings of the 4th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 42–51, Seattle, Washington. Association for Computational Linguistics.

Christopher Haberland and Shane Steinert-Threlkeld. 2025. Quantifiers that are more monotone are easier to learn. In *Proceedings of Semantics and Linguistic Theory (SALT 35)*, volume 35.

Nathaniel Imel, Qingxia Guo, and Shane Steinert-Threlkeld. 2024. An Efficient Communication Analysis of Modal Typology. LingBuzz Published In:.

Charles Kemp and Terry Regier. 2012. Kinship categories across languages reflect general communicative principles. *Science (New York, N.Y.)*, 336(6084):1049–1054.

Charles Kemp, Yang Xu, and Terry Regier. 2018. Semantic typology and efficient communication. *Annual Review of Linguistics*, pages 1–23.

Mora Maldonado and Jennifer Culbertson. 2019. Something about \emph{us}: Learning first person pronoun systems. In *Proceedings of the 41st Annual Meeting of the Cognitive Science Society (CogSci 2019)*.

Mora Maldonado, Jennifer Culbertson, and Wataru Uegaki. 2022. Learnability and constraints on the semantics of clause-embedding predicates. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 44(44).

Steven T. Piantadosi. 2014. LOTlib: Learning and Inference in the Language of Thought. Published: available from https://github.com/piantado/LOTlib.

Jake Quilty-Dunn, Nicolas Porot, and Eric Mandelbaum. 2022. The Best Game in Town: The Re-Emergence of the Language of Thought Hypothesis Across the Cognitive Sciences. *Behavioral and Brain Sciences*, pages 1–55.

Shane Steinert-Threlkeld. 2020. An Explanation of the Veridical Uniformity Universal. *Journal of Semantics*, 37(1):129–144.

Shane Steinert-Threlkeld. 2021. Quantifiers in Natural Language: Efficient Communication and Degrees of Semantic Universals. *Entropy. An International and Interdisciplinary Journal of Entropy and Information Studies*, 23(10):1335.

Shane Steinert-Threlkeld, Nathaniel Imel, and Qingxia Guo. 2023. A semantic universal for modality. *Semantics and Pragmatics*, 16:1:1–20.

Shane Steinert-Threlkeld and Jakub Szymanik. 2019. Learnability and semantic universals. *Semantics and Pragmatics*, 12:4:1–39.

Shane Steinert-Threlkeld and Jakub Szymanik. 2020. Ease of learning explains semantic universals. *Cognition*, 195:104076.

David Strohmaier and Simon Wimmer. 2022. Contrafactives and Learnability. *Proceedings of the Amsterdam Colloquium*, pages 305–312.

Wataru Uegaki. 2021. The informativeness / complexity trade-off in the domain of Boolean connectives. *Linguistic Inquiry*.

Mind the Gap: Computational Quality Assurance of Crowd-Sourced Linguistic Knowledge on Latin and Italian Morphological Gaps

Jonathan Sakunkoo and **Annabella Sakunkoo**
Stanford University OHS
{jonkoo,apianist}@ohs.stanford.edu

1 Introduction

> The past tense of "forgo" is forwent. So, you would say: "I forwent this position." It's a bit formal or uncommon in modern usage, but grammatically correct.

Above is a response from GPT-4o when asked what the past tense for "forgo" is. Yet, most fluent English speakers would find *forwent* unnatural, ineffable (Gorman, 2023), and unacceptable (Embick and Marantz, 2008). Most English speakers would also be unable to find the right, natural form for the past tense of *forgo* (Gorman and Yang, 2019). Words such as *forgo* are instances of defective verbs or morphological gaps in which expected forms are absent—a problematic intrusion of morphological idiosyncrasy (Baerman and Corbett, 2010).

While inflectional gaps are not a recently discovered phenomenon, they "remain poorly understood" (Baerman and Corbett, 2010) and documenting them requires extensive human expertise and effort. For scarce linguistic phenomena in less-studied languages, Wikipedia and Wiktionary serve as among the few widely accessible and frequently utilized resources, consistently ranked among the most popular websites globally. With its extensive reach and usage, crowd-sourced content is a potentially valuable but underexplored resource although its user-contributed nature has sparked controversy on its overall trustworthiness.

In this study, we conduct computational analyses of inflectional gaps by customizing UDTube (Yakubov et al., 2024), a scalable state-of-the-art neural morphological analyzer trained with Universal Dependencies (a collection of corpora of morphologically annotated text in different languages), to incorporate mBERT (Devlin et al., 2019) as an encoder and annotate large corpora of text in Latin and Italian (Conneau et al., 2020). The resulting massive annotated data are then used to measure the frequency of certain inflectional forms of interest and validate lists of defective verbs scraped and compiled from Wiktionary's Latin and Italian pages to verify which verbs are confirmed computationally to be inflectional gaps.

By bridging computational techniques with linguistic analysis, the study contributes to linguistics of less-explored languages and offers novel insights and computational methodologies for scalable quality assurance and validation of crowd-sourced content, while addressing gaps in linguistic knowledge.

2 Data

This study uses Universal Dependencies (UD), Common Crawl, and Wiktionary in the computational validation of morphological gaps. Universal Dependencies is a collection of multilingual treebanks for syntactic and morphological analysis across languages (Nivre et al., 2017). We utilize the largest available treebanks for Italian and Latin in the UD dataset. For corpora, we use an 8.3GB dataset containing approximately 5 billion tokens of diverse Italian text and a 640MB dataset with approximately 390 million tokens of Latin text.

3 Methods

As shown in Figure 1, this study uses a computational approach to validate inflectional gaps in Latin and Italian in three major steps: (1) Training UDTube with Universal Dependencies, (2) Annotating Large-Scale Text with UDTube[1], and (3) Validating Defective Forms.

4 Results and Conclusion

In the evaluation of defective lemmata listed in Wiktionary against corpus evidence, lemmata are classified into likely defective (based on expert-recommended frequency threshold of 10), on the edge, and likely not defective.

[1] The tuned morphological analyzer achieves 98% and 96% accuracy on the Latin and Italian test sets, respectively.

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 402-403.
Eugene, Oregon, July 18-20, 2025

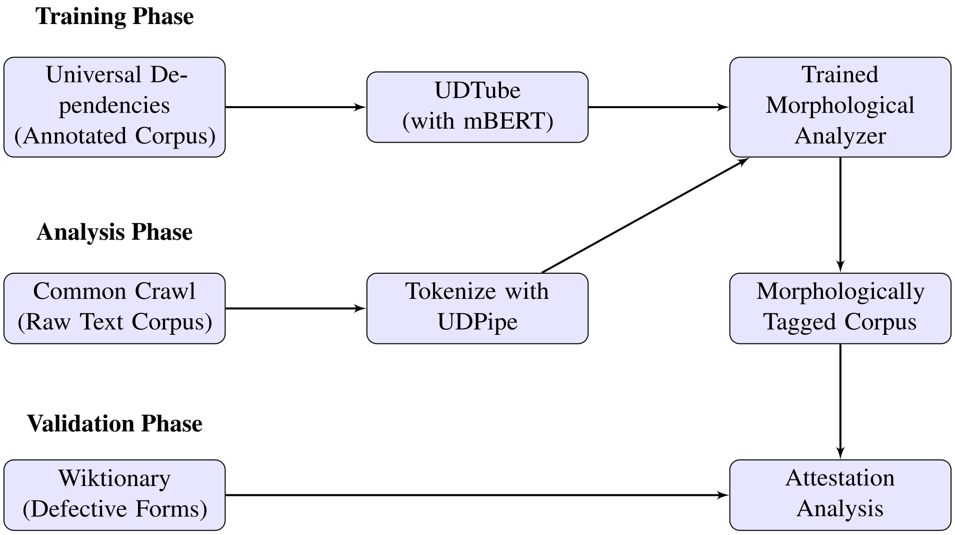

Training Phase
Universal Dependencies (Annotated Corpus)
UDTube (with mBERT)
Trained Morphological Analyzer
Analysis Phase
Common Crawl (Raw Text Corpus)
Tokenize with UDPipe
Morphologically Tagged Corpus
Validation Phase
Wiktionary (Defective Forms)
Attestation Analysis

Figure 1: Workflow for computational validation of morphological gaps, using UDTube

Occurrences	Latin	Italian
Likely defective: ≤ 10	67.4%	79%
On the edge: $11 - 100$	25.4%	17%
Likely not defective: > 100	7.2%	4%

Table 1: Summary of defective forms in Wiktionary

Based on this result, Wiktionary's list of defective verbs in Italian is 1.8 times less likely to contain errors compared to Latin. The computational results, together with manual verification by human experts, suggest that while Wiktionary provides a reliable account of Italian morphological gaps, at least 7% of Latin lemmata listed as defective are unlikely to be truly defective. This discrepancy highlights potential limitations of crowd-sourced wikis as definitive sources of linguistic knowledge, particularly for less-studied phenomena and languages, despite their value as resources for rare linguistic features. This study presents a novel computational approach to validating defectivity in a crowd-sourced linguistic resource and contributes to expanding our morphological knowledge.

5 Acknowledgement

We are grateful to Kyle Gorman for valuable advice and to Yale NENLP researchers, Dan Jurafsky, and reviewers for insightful feedback for future work.

References

Matthew Baerman and Greville G. Corbett. 2010. *Defective Paradigms: Missing Forms and What They Tell Us*. Oxford University Press, Oxford.

Jeremy K. Boyd and Adele E. Goldberg. 2011. Learning what not to say: The role of statistical preemption and categorization in a-adjective production. *Language*, 87(1):55–83.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

David Embick and Alec Marantz. 2008. Architecture and blocking. *Linguistic Inquiry*, 39(1):1–53.

Kyle Gorman. 2023. *Morphological Defectivity*.

Kyle Gorman and Charles Yang. 2019. *When Nobody Wins*. Springer International Publishing.

Joakim Nivre, Daniel Zeman, Filip Ginter, and Francis Tyers. 2017. Universal Dependencies. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, Valencia, Spain. Association for Computational Linguistics.

Daniel Yakubov, Kyle Gorman, and Github Contributor Jonathan Sakunkoo. 2024. UDTube: A tool for universal dependency-based linguistic analysis.