SCiL 2026

Society for Computation in Linguistics (2026)

Proceedings of the Conference

July 3-4, 2026

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

With thanks to

©2026 Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

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

ISBN 979-8-89176-412-5

SCiL 2026 Editors' Note

Rob Voigt, Alex Warstadt, Naomi Feldman, Tal Linzen

This volume contains research presented at the ninth annual meeting of the Society for Computation in Linguistics (SCiL), held as a workshop co-located with the Annual Meeting of the Association for Computational Linguistics in San Diego, California, July 3-4, 2026.

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 86 main conference submissions, of which 19 were selected for oral presentation ($\tilde{2}2\%$) and 38 for poster presentation ($\tilde{4}4\%$).

To build connections between the ACL and SCiL communities, we organized two crossover events. First, at SCiL itself we invited submissions of accepted ACL Main Conference and Findings papers with strong linguistic relevance to join for poster presentation. We received 7 submissions via this mechanism and all were accepted. Second, we held a linguistics-oriented symposium during the ACL main conference entitled "Linguistics and NLP in the LLM Era." This took the form of a 90-minute panel with four panelists with a strong history of participation in both the ACL and SCiL communities: Allyson Ettinger, Richard Futrell, Zoey Liu and Tom McCoy.

Further information can be found at our website: https://sites.google.com/view/scil2026.

We thank our reviewers for their indispensable help in selecting the research for presentation at the conference. Thanks also to Joe Pater and Brandon Prickett for their help with publishing the proceedings.

We are deeply grateful to the U.S. National Science Foundation for providing a grant of $30,000 to enable cross fertilization between the ACL-SCiL communities, which allowed us to provide significant scholarships towards the costs of attendance for SCiL authors.

Organizing Committee

General Chair

Rob Voigt, University of California, Davis

Local Chair

Alex Warstadt, University of California, San Diego

Sponsorship Chairs

Naomi Feldman, University of Maryland
Tal Linzen, New York University

Program Committee

Reviewers

Jiwon Yun, State University of New York at Stony Brook
Christabel Acquaye, University of Maryland, College Park
Mai Al-Khatib, University of Minnesota - Twin Cities
Suhas Arehalli, Macalester College
Barend Beekhuizen, University of Toronto
Caleb Belth, University of Utah
Shohini Bhattasali, University of Toronto
Veronica Boyce, Massachusetts Institute of Technology
Canaan Breiss, University of Southern California
Jonathan Brennan, University of Michigan - Ann Arbor
Dylan Bumford, University of California, Los Angeles
Alexander Clark, University of Gothenburg
Lindy Comstock, University of California, Los Angeles
Aniello De Santo, University of Utah
Brian Dillon, University of Massachusetts at Amherst
Karthik Durvasula, Michigan State University
Micha Elsner, Ohio State University
Anna Feldman, Montclair State University
Richard Futrell, University of California, Irvine
Luke Gessler, Indiana University
Matthew Goldrick, Northwestern University, Brown University and Johns Hopkins University
Thomas Graf, State University of New York, Stony Brook
Philippe de Groote, INRIA
Kenneth Hanson, Yale University
Bruce Hayes, University of California, Los Angeles
Kasia Hitczenko, University of Delaware
Katherine Howitt, University of Maryland, College Park
Lindsay Hracs, University of Calgary
Nick Huang, National University of Singapore
Tim Hunter, University of California, Los Angeles
Cassandra L Jacobs, State University of New York, Buffalo
Adam Jardine, Rutgers University
Gaja Jarosz, University of Massachusetts at Amherst
Tamar Johnson, University of Amsterdam
Vsevolod Kapatsinski, University of Oregon
Itamar Kastner, University of Edinburgh, University of Edinburgh
Andrew Kehler, University of California, San Diego
Maayan Keshev, Hebrew University of Jerusalem
Dongsung Kim, Ewha Womans University
Tracy Holloway King, Adobe Systems
Christo Kirov, Google
Gregory M Kobele, Universität Leipzig
Hahn Koo, San Jose State University
Andrew Lamont, University College London, University of London
D Terence Langendoen, University of Arizona
Sarah Ita Levitan, CUNY Hunter College

Constantine Lignos, Brandeis University
Zoey Liu, University of Florida
Yang Janet Liu, University of Pittsburgh
Ling Liu, Amazon
Giorgio Magri, CNRS
Fred Mailhot, Dialpad, Inc.
Robert Malouf, San Diego State University
Connor Mayer, University of California, Irvine
Paola Merlo, Idiap Research Institute and University of Geneva, Switzerland
Timothee Mickus, University of Helsinki
Sarah Moeller, University of Florida
David R. Mortensen, Carnegie Mellon University
Sathvik Nair, University of Maryland
Scott Nelson, University of Illinois Urbana-Champaign
Ifeoma Okoh, University of Rochester
Yohei Oseki, National Institute of Informatics and University of Tokyo
Joe Pater, University of Massachusetts Amherst
Lisa Pearl, University of California, Irvine
Laurel Perkins, University of California, Los Angeles
Tiago Pimentel, ETH Zurich
Brandon Prickett, University of Massachusetts Amherst
Eric Raimy, University of Wisconsin - Madison
Kyle Rawlins, Johns Hopkins University
Jonathan Rawski, San Jose State University
Mats Rooth, Cornell University
Ollie Sayeed, University of Pennsylvania
Sebastian Schuster, University of Vienna
Gyu-Ho Shin, University of Illinois Chicago
Morgan Sonderegger, McGill University
Shane Steinert-Threlkeld, University of Washington
Michelle Suijkerbuijk, Radboud University
Craig Thorburn, University of Chicago
Simon Todd, University of California, Santa Barbara
Abdullah Topraksoy, Istanbul University
Shiva Upadhye, University of California, Irvine
Géraldine Walther, George Mason University
Sheng-Fu Wang, Academia Sinica
Bonnie Webber, Edinburgh University, University of Edinburgh
Shira Wein, Amherst College
Ethan Gotlieb Wilcox, Georgetown University
Agata Wolna, Massachusetts Institute of Technology
Qihui Xu, Ohio State University
Olga Zamaraeva, Universidad de La Coruña

Table of Contents

Measuring Perceptions of Personhood with Semantic Proto-role Properties
Elizabeth Spaulding Hoefer and James Martin 1

Small Neural Networks as Models of Cross-Linguistic Speech Perception
Annika Shankwitz 15

A Family of Effective Methods for Decompiling Canonical Acceptors, Instantiated for Languages of Dot-Depth One and Tier-Based Extensions
Dakotah Lambert 25

Examining the Limits of Word2Vec with Toki Pona
Daniel Huang and Hongchen Wu 37

Learning Latent Representations with Progressive Hypothesis Space Expansion
Jonathan Charles Paramore 47

What Exactly do Children Receive in Language Acquisition? A Case Study on CHILDES with Automated Detection of Filler-Gap Dependencies
Zhenghao Zhou, William Dai, Maya Viswanathan, Simon Charlow, R. Thomas McCoy and Robert Frank 57

Determinants of Hesitations and Repetitions in Hindi Spontaneous Speech
Eashani Sharma, Ishita Arun and Samar Husain 59

An LLM Investigation into Inherent and Structural Case Representation: a German Case Study
Iona Carslaw, András Bárány, Itamar Kastner and Mark Steedman 72

A Logical Analysis of Autosegmental Approaches to Root-and-Pattern Morphology in Arabic
Sarah Payne 90

Comonadic Morphophonology: A Compositional Framework for Context-Dependent Morphological Rules in Finnish
Yongseok Jang 100

The Development of Spectral and Temporal Encodings in Speech Sounds
Frank Lihui Tan and Youngah Do 113

The Spanish Learner and Heritage Speaker Dependency Treebank
Valeria Pagliai, Sergio José Salazar Rodó, Emiliana Pulido, Andres Gutierrez-Quintero and Zoey Liu 127

Omnivorous Agreement, like Uyghur Backness Harmony, is a Challenge for Tier-Based Strict Locality
Allison Verbil and Tim Hunter 129

Modelling the Diachronic Emergence of Phoneme Frequency Distributions
Fermin Moscoso Del Prado Martin and Suchir Salhan 138

Morpheme structure phonotactics: a categorical model for morpho-phonological productivity in Russian vowel-zero alternations
Daniar Kasenov 147

Quantifying the cross-linguistic effects of syncretism on agreement attraction
Utku Turk and Eva Neu 160

Compiling Search & Change Rules into Subsequential Finite-State Transducers
Malek Azadegan 171

What Matters in Tonotactic Learning
Han Li and Jeffrey Heinz 180

Investigating Syntactic Biases in Multilingual Transformers with RC Attachment Ambiguities in Italian and English
Michael Kamerath and Aniello De Santo 191

A Feature-Driven Tensor Semantics for Minimalist Grammars
John Paulson, Aniello De Santo and Jonathan Rawski 219

Word Predictability on Code-switching Points in Cantonese–English Discourse
Ariel Shuk Ling Chan, Yanting Li and Jacob Poschl 230

Non-literal Meaning Representation in the Brain during Naturalistic Listening
Zhengwu Ma, Yuhan Huang, Chengcheng Wang and Jixing Li 244

Probing the Attention Representation of Filler-Gap Dependency in Transformers
Ruoqing Yao and Pranav Anand 258

Learning Stress in Arabic Low-Resource Settings
Abed Qaddoumi, Jordan Kodner, Owen Rambow, Salam Khalifa and Jeffrey Heinz 262

Rating-based property axes: Agentivity and split intransitivity
Eva Neu, Brian Dillon and Katrin Erk 280

Mapping the meaning of Hungarian impulsative constructions
Ágnes Kalivoda, Robert Malouf and fackerman@ucsd.edu fackerman@ucsd.edu 291

Efficient universal generation in a fragment of Optimality Theory
Paul Siewert 294

Learning Process Interaction Through Simplex ISL Transducers
Chenli Wang and Adam Jardine 304

Do I know what I want to say? Modeling meaning uncertainty in RSA
Anzi Wang, Carolyn Jane Anderson and Grusha Prasad 313

Learning tone sandhis in Structural Optimality
Izabel Ilie, Andrew Lamont and Brandon Prickett 329

Making "concreteness" more concrete
Yanting Li, Gregory Scontras and Richard Futrell 338

Frequency modulates structural choice in Turkish suspended affixation: a latent-process account
Utku Turk, Eva Neu, Özge Bakay, Brian Dillon and Gaja Jarosz 343

Effect of case markers during agreement production: A model comparison using Armenian forced choice data
Pranab Bagartti and Samar Husain 353

Propositional compositionality in neural language models
Jane Li, Abhinav Patil and Kyle Rawlins 365

Can language models process linguistic deference?
Youngdong Cho and Chloe Dokyung Kwon 369

CrosSing: Cross-Scale Reasoning Evaluation on LLMs against Humans
Qi Han, Yifan Wu and Marten Van Schijndel 379

Fine-tuning Whisper Across 81 Languages
Shivam Singh and Alex Warstadt 408

A Transparent Model of Syntactic and Semantic Cue-based Retrieval
Shisen Yue and John T. Hale 411

How much capacity does Turkish inflection require? An empirical study of GRU encoder–decoder bottlenecks.
Fred Mailhot 423

The signal is coming from inside the noun phrase! Tracking semantic proto-role inferences during sentence processing
Lucas Y. Li, Zander Lynch and Marten Van Schijndel 427

Quantifying mutual intelligibility gradients in Turkic languages using language models
Moldir Baidildinova, Shiva Upadhye, Austin Wagner, Connor Mayer and Richard Futrell 442

Algebraic Classification of Reduplicative Processes
Matthew Hayden 447

Learning reduplicative templates as hidden structures: the case of reduplication-phonology interactions
Yang Wang 460

Do Large Language Models Acquire Phrase-Based Processing? Evidence from Eye Movements and Model-Brain Alignment After Fine-Tuning
Xufeng Duan, Zhengwu Ma, Zhaoqian Yao, Jixing Li and Zhenguang Cai 464

Roles of Predictability and Acoustic Distance in Sound Discrimination via Contrastive Learning
Shuhao Zhang and Youngah Do 477

Graded Expectations: Do Large Language Models Show Human-like Sensitivity to the Likelihood of Deceptive Speech Acts?
Xingyuan Zhao and Seana Coulson 487

Lexical exceptionality in paradigm-specific learning: modeling stem-final obstruent alternations in Korean verbs and adjectives
Stella Eunsoo Hong 496

Adaptive Speech Perception: Empirical Indeterminacy and a Path Forward
Shawn N. Cummings, T. Florian Jaeger, Chigusa Kurumada and Xin Xie 513

Modeling generalization in perceptual learning of speech
Yiming Lu, Xinyu Leslie Liao, Alejandro Tabas and Xin Xie 529

Output Languages of Strictly Local Phonological Processes
Anton Hampe 542

Phonological Processes as Modal Transductions
Tatevik Yolyan and Jesse Comer 549

Measuring Perceptions of Personhood with Semantic Proto-role Properties

Elizabeth Spaulding Hoefer
University of St. Thomas
elizabeth.hoefer@stthomas.edu

James Martin
University of Colorado Boulder
james.martin@colorado.edu

Abstract

We show that semantic proto-role properties can be used as a tool to measure implicit human perceptions of *agency* and *patiency* of entities in human-generated text. First, we demonstrate that silver-generated semantic proto-role property labels are strongly correlated with both human judgment and a probabilistic text-based measure of anthropomorphism. Then, we use our measure to quantify linguistic idiosyncrasies across different AI-related Reddit communities. Our measure shows that subreddits dedicated to discussing AI companionship ascribe higher sentience to "bots" and higher agency to "companies" when compared to other subreddits. This phenomenon reveals not only the unique way in which chatbots are anthropomorphized in such subreddits, but also the users' keen awareness of their power imbalance with the companies that created the chatbots.

1 Introduction

It has long been recognized that the semantics of ordinary human language reflects the attitudes and biases underlying the human culture that produced it. For instance, metaphors are not only descriptive, but can change the way people think by shaping our understanding of novel concepts (Lakoff and Johnson, 2008; Lakoff and Turner, 2009; Landau et al., 2010). Thus, the use of metaphor in human speech or text can reveal implicit attitudes of the speaker (Ruscher, 2017).

Research in computational semantics, corpus linguistics, and machine learning has shown that machines trained on text corpora adopt the implicit human attitudes contained within them. For instance, Caliskan et al. (2017) uncover a wide spectrum of human-like biases in word embeddings using Implicit Association Tests (Greenwald et al., 1998), such as associations of *insects* with *unpleasantness*, or stronger associations of female names with family words than with career words compared to male names. Similarly, Ash et al. (2023) use word embeddings to quantify human perceptions of *agency* and *patiency* of 255 entities, and track changing perceptions of women and animals over several decades of historical text. Cheng et al. (2024) quantify the anthropomorphism of AI entities in text with a machine learning metric called AnthroScore, finding that news headlines anthropomorphize AI more than research papers.

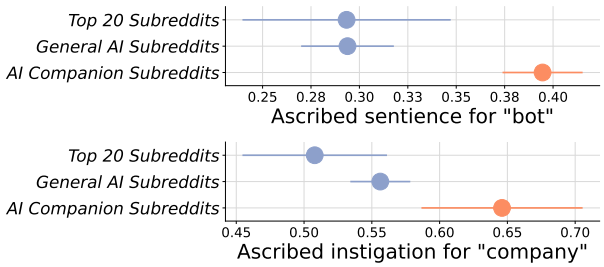


Figure 1: Subreddits that discuss AI companionship ascribe higher *sentience* to *bots*, and higher *instigation* to *companies*, compared to other subreddits.

We show that semantic proto-role labeling (SPRL), a task which classifies and characterizes the predicates (events) and arguments (entities) of a sentence, can also be used to measure how much the language in a corpus anthropomorphizes entities. Furthermore, we show that SPRL can quantify and separate differing dimensions of personhood. As a case study, we apply our method to study the perception of AI-related entities (such as chatbots and LLMs) across different Reddit communities. We find that, compared to other communities, communities that discuss the usage of AI companions ascribe higher levels of sentience to entities such as *bot*, but lower levels of sentience to entities such as *LLM* and *GPT*, suggesting a layered conceptualization of AI. We also find that AI companion users ascribe higher agency to *companies*, which reflects their concern with the level of control the companies hold over their companions.

Replication package available here.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 1–14
July 3-4, 2026 ©2026 Association for Computational Linguistics

2 Related work

2.1 Defining personhood

Ash et al. (2023) describe a two-dimensional view of personhood comprised of mental properties of *agency* (the ability to act, plan, and decide) and *experience/patiency* (the ability to feel, hunger, and desire; Baker 2000; Gray et al. 2007), which materializes in law, and motivates much of the way we reason about morality (Freeland, 1985; Shaver, 1985; Weiner, 1995). Using a different model of mind perception, Cheng et al. (2024) define anthropomorphism by citing a three-component model of agency itself (the "ABC model"), which studies the role of agency in dehumanizing metaphors: "Full agents possess the ability to (1) experience emotion and feel pain (affective mental states), (2) act and produce an effect on their environment (behavioral potential), and (3) think and hold beliefs (cognitive mental states)." (Tipler and Ruscher, 2014) These are the "human-like qualities" that can be ascribed to non-human entities, and can be measured by AnthroScore.

Anthropomorphizing AI Personifying or anthropomorphizing inanimate objects—that is, attributing human characteristics to nonhuman things, or viewing a nonhuman object through a social lens—is a normal human tendency (Zimmerman et al., 2024). Today, people increasingly perceive AI as more warm, competent, and human-like (Cheng et al., 2026). An individual's beliefs and perceptions of AI technology influence their willingness to use it (Kelly et al., 2023), and, in turn, increased usage and familiarity with chatbots is associated with an increased likelihood to attribute some form of phenomenal consciousness to the chatbot (Colombatto and Fleming, 2024), and users who perceive chatbots as more human-like and conscious are more likely to report positive opinions of the chatbot (Guingrich and Graziano, 2025). However, anthropomorphizing AI carries significant risks, such as encouraging overreliance on system outputs (Abercrombie et al., 2023; Chiesurin et al., 2023), including overly cautious, sycophantic, or incorrect health advice in therapeutic contexts (Grabb et al., 2024; Jung et al., 2025), and encouraging emotional relationships with AI systems. Some argue that such relationships can detract from human relationships, and can be exploited by the companies that control the systems (Zimmerman et al., 2024); one study analyzed several thousand chatbot conversations of users of the AI companion app Replika and identified six harms: relational transgression, harassment, verbal abuse, self-harm, misinformation, and privacy violations (Zhang et al., 2025). Other research has found that users of Replika can project gender stereotypes onto the bot (Depounti et al., 2023), which mirrors and affirms such stereotypes via sycophantic responses (Lima and Belk, 2025).

AnthroScore Researchers are increasingly interested in quantifying the anthropomorphism of AI because of the potential risks and harms. As such, Cheng et al. (2024) develop a language measure to quantify anthropomorphism in text using RoBERTa (Liu et al., 2020). For a masked entity e, AnthroScore is the log of the ratio of the probability that a human pronoun replaces the mask, over the probability that a non-human pronoun replaces the mask. So, a higher AnthroScore means greater anthromorphism of the masked entity e in the sentence: $Anth(e) = \log \frac{P_{HUMAN}(e)}{P_{NON-HUMAN}(e)}$. More details on AnthroScore can be found in Appendix B.

Human judgement survey Earlier work has sought to characterize people's understanding of personhood or *mind* of various entities (Gray et al., 2007; Gray and Wegner, 2009). We utilize the data from Ash et al. (2023), who collected human judgements ($N = 3,181$ respondents) on 255 common entities in the English language. The researchers collected implicit judgements on the agency and experience of these entities (see survey questions in Table 4). They then average the responses in each category into two scores for agency $a_H(e)$ and patiency/experience $p_H(e)$, with $e \in E$ representing each entity and H for "human survey." We utilize this data as it is the largest validated set of entities with human judgements of agency and experience to our knowledge.

2.2 Semantic proto-roles

Because *agency* and *experience/patiency* define personhood, and thus, anthropomorphism, we turn to a linguistic theory of agency and patiency to help us quantify humanizing and anthropomorphizing language. Entities within sentences can be classified as more agent-like or more patient-like[1] on the basis of a set of properties, introduced by Dowty (1991): the properties of a prototypical agent ("proto-agent") are *sentience*, *awareness* of the event described by the verb, *independent exis-*

[1]In semantics, a "patient" is the entity that experiences the effects of the action in the sentence.

tence of the event, *volition* with respect to the event, and *instigation* of the event. Semantic proto-role labeling (SPRL) is the computational task which assigns these properties to entities within sentence (Reisinger et al. 2015; White et al. 2016; Rudinger et al. 2018; see Appendix A for further explanation of this task). For example, in the sentence "The cat sat on the mat", a SPRL parser would label *the cat* with:

{ +volition, +sentience, +awareness, +independent existence, +instigation }

while *the mat* would be labeled with:

{ –volition, –sentience, –awareness, +independent existence, –instigation }.

Thus, by Dowty's theory, *the cat*, which exhibits more proto-agent properties, is the agent of the sentence, while *the mat* is the patient.

3 Experiments

We address the following questions:

(RQ1) Can we learn about human perceptions of personhood of entities of interest by performing semantic proto-role labeling on text?

  (a) Can semantic proto-role properties quantify differing dimensions of personhood in text?

(RQ2) Do users of AI chatbot companions perceive the agency and experience of chatbots differently than users of AI chatbots for other reasons?

To answer RQ1, we validate our SPRL measures on both human judgement from Ash et al. (2023), and the probabilistic, text-based anthropomorphism measure from Cheng et al. (2024). To answer RQ2, we analyze the text of user groups taken from Reddit using our validated SPRL measure.

3.1 Data for validating SPRL measures

COHA We utilize the Corpus of Historical American English (COHA; Davies, 2021) for best comparison to Ash et al. (2023). COHA is a large corpus consisting of English language usage across several decades, starting in the 1800s. The sample we used consists of 3.6 million words and over 200,000 sentences. After filtering for noise, length, and the presence of entities with human judgements, 47,301 sentences remained, over which 18,781 predicate-argument pairs were identified by the SPRL system. Further filtering details are in Appendix C.1.

For the analysis on COHA, we utilize the same 200+ entities chosen by Ash et al., based on previous work in social perceptions of agency and patiency. Because many of the entities in the full list are infrequently mentioned in COHA, we additionally separate entities into subsets of different sizes $E_{N \geq n}$. Membership in $E_{N \geq n}$ is determined by the number of silver[2] SPRL mentions across the COHA sample n, so that the subset $E_{N \geq 100}$ contains all entities which have at least 100 mentions in the silver-labeled data. Additionally, Ash et al. stipulate a pre-registered subset of entities designed to be representative of different agency and experience concepts, which we call E_{pre}. We evaluate on this subset as an additional validation of our measure, but our main analysis is over $E_{N \geq 100}$ and $E_{N \geq 5}$, which contain common entities such as *woman*, *girl*, *god*, *army*, *dad*, *fish*, etc. The complete list of 200+ entities with human judgement scores, along with our specific subsets, can be found in Table 11.

ACL Abstracts As in Cheng et al. (2024), we analyze abstracts taken from the ACL Anthology[3], which contains >300,000 computational linguistics papers. We parse all ACL abstracts and manually review the identified arguments, sorted by number of mentions throughout the corpus. We select 88 frequent entities, ensuring we cover all entities covered in Cheng et al. (2024), and then filter sentences as in COHA, over our selected entities, and end up with 131,349 sentences which contain 224,791 mentions with silver SPRL. As in the COHA analysis, we separate entities into subsets $E_{N \geq n}$, where entities have at least n silver SPR labels. We show entity lists in Table 12.

3.1.1 Reddit data

Reddit, a massive online discussion forum with thousands of different communities ("subreddits"), is often used as a source of observational data in social science and digital humanities research (Proferes et al., 2021). Each subreddit has its own culture and set of norms, and the ability to remain semi-anonymous on Reddit means that users can discuss controversial or stigmatizing topics relatively freely. Because we are interested in the linguistic signals in the discussions among users of AI chatbot companions—a topic that is considered

[2]"Silver-labeled" data is data that has been automatically annotated by a trained model rather than manually annotated by a human (which we refer to as "gold-labeled" data).

[3]https://aclanthology.org/

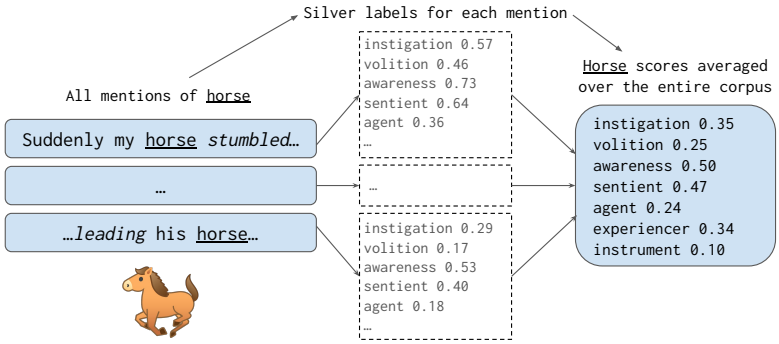


Figure 2: The process for scoring an entity *horse*.

stigmatizing (Dubé et al., 2023)—we chose Reddit as our data source to investigate RQ2.

We scraped text from 38 different subreddits which we classified under three umbrellas to capture three different user groups: (1) subreddits for individuals who use AI chatbot companions; (2) subreddits for general AI users, i.e. individuals who use chatbots for miscellaneous, but mainly productivity-related reasons; and (3) the top 20 most popular subreddits, for a more general population of people who may or may not use chatbots at all. After filtering for our entities and for length, we obtained 17,184 sentences for the AI companion category, 63,331 sentences for the general AI category, and 11,469 sentences for the top 20 category. See Table 9 for a full list of the subreddits.

Ethical considerations This study was approved by our institution's institutional review board under exempt status due to Reddit's public nature, but because of the sensitive nature of some of the topics discussed in our data, we take extra steps to protect the privacy of the users. Because it is relatively easy to trace a particular Reddit user to an exact quotation using a search engine, we follow previously established guidelines in the field and do not release raw data or include direct quotations from any Reddit users, and instead use bricolage-style paraphrases to inhibit traceability (Markham, 2012; Sanches et al., 2019).

3.2 Methodology

First, to establish a correlation between the H measure (the survey judgments from Ash et al.) and semantic proto-role labels, we analyze entities in text using both gold (annotated) and silver (system-produced) semantic proto-role labels. To produce silver SPR labels, we reproduce the full semantic proto-role labeling system from Spaulding et al. (2023). Our system achieves a micro-F1 of 83.2 on SPR properties. Details on our system, including individual F1s on different properties and components, can be found in Appendix A.

AGENT	EXPERIENCER	INSTRUMENT
+instigation	-instigation	+instigation
+volition	-volition	-volition
+awareness	+awareness	-awareness
+sentient	+sentient	-sentient
+change loc		+change loc
+ind. existence		

Table 1: All SPRL categories used for our measure, with binary properties for each one.

3.2.1 SPRL Measures

We configure a number of SPRL categories based off of Dowty's "cluster-concepts" for the thematic roles of Agent, Experiencer, and Instrument (Dowty, 1991, p. 578). Table 1 shows these categories. In addition to these categories, we also score entities on four binary semantic proto-role properties alone: instigation, volition, awareness and sentience (see Appendix A.1 for a full list).

For the SPR labels alone, we score entities by averaging the *probability output* from the SPRL parser's logits. For the three cluster-concepts from Table 1, we score in two different ways. First, we score an argument for a cluster by summing the raw probability output for each of the *positive* constituent properties if and only if the probability output is greater than 0.5. In other words, for an entity mention e with SPRL probabilities p_s, we score the EXPERIENCER cluster as follows:

$$\text{EXPERIENCER}(e) = p_{\text{aware}} * [\![p_{\text{aware}} > 0.5]\!] + p_{\text{sentient}} * [\![p_{\text{sentient}} > 0.5]\!]$$

Because this method does not take into account the negative properties in each cluster, we score by awarding a 1 for each constituent property that matches exactly, so for an entity mention e with SPRL probabilities p_s, we generate the strict score for the INSTRUMENT cluster as follows:

$$\begin{aligned}\text{STRICT INSTRUMENT}(e) = 1 * [\![p_{\text{instigation}} > 0.5]\!] \\ +1 * [\![p_{\text{volition}} \leq 0.5]\!] + 1 * [\![p_{\text{aware}} \leq 0.5]\!] \\ +1 * [\![p_{\text{sentient}} \leq 0.5]\!] + 1 * [\![p_{\text{chg loc}} > 0.5]\!]\end{aligned}$$

For each SPRL measure, and each entity, we produce a score for that entity, as shown in Figure 2 for the entity *horse*. To validate our SPRL measures, we gather entity scores across the COHA sample and calculate both Pearson's and Spearman's correlation coefficients between each H measure and each SPRL measure. We do the same thing across the ACL abstracts and AnthroScore. We also calculate and report the associated p-value, and consider our results statistically significant at $p < 0.01$.

4 Results and discussion

4.1 RQ1: SPRL measures correlate with human judgements

We find that our SPRL measures correlate with the human judgements on the entities collected in Ash et al. (2023). We compute our scores across a number of different subsets of entities and two datasets to check for robustness.

SPR1&2 and H Measures First, we run an initial pilot analysis and aggregate the measures across the *gold* SPRL annotations, two datasets called SPR1 and SPR2 (see Appendix A). This verification step checks that the human judgements of the SPRL annotations (i.e., the data that the SPRL parser was trained on) align with the human judgements from the survey data. We completed this step before generating the silver labels to verify that the two sources of human judgements were comparable. All ρ coefficients calculated on E_{gold} in SPR1 and SPR2 can be found in Table 5 in Appendix C.2. While the subset of entities we had enough data to measure is small, with $|E_{gold}| = 17$, we found that the gold SPRL aggregates correlated highly with the human survey scores: Our AGENT measure had Pearson's $\rho = 0.78$ with $a_H(E_{gold})$ (the human survey aggregate for the agency dimension of personhood), and our EXPERIENCER measure had $\rho = 0.814$ with $p_H(E_{gold})$, with $p < 0.01$.

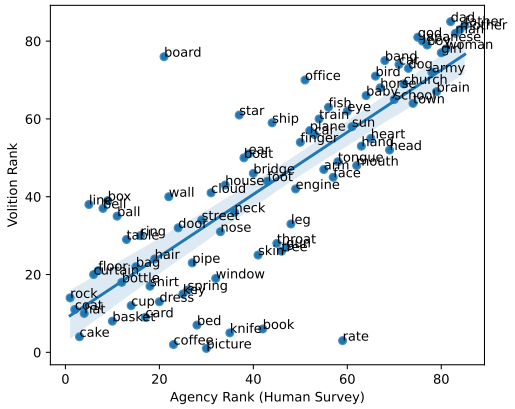
H Measure	SPRL Measure	Pearson's ρ	Pearson's p	Spearman's ρ	Spearman's p
agency (a_H)	AGENT	0.922	0.000	0.823	0.000
	instigation	0.923	0.000	0.852	0.000
	volition	0.933	0.000	0.821	0.000
	awareness	0.929	0.000	0.837	0.000
	sentient	0.928	0.000	0.844	0.000
	change of location	0.269	0.102	0.296	0.071
	independent existence	0.756	0.000	0.796	0.000
patiency (p_H)	EXPERIENCER	0.901	0.000	0.733	0.000
	awareness	0.901	0.000	0.745	0.000
	sentient	0.909	0.000	0.763	0.000
$a_H - p_H$	AGENT - EXPERIENCER	0.710	0.000	0.494	0.001

Table 2: ρ coefficients for a_H, p_H, and d_H and SPRL measures over the subset $E_{N\geq 100}$ wherein each entity has at least 100 mentions. ($|E_{N\geq 100}| =$ 38 entities total). Text in CAPS denotes clusters from Table 1.

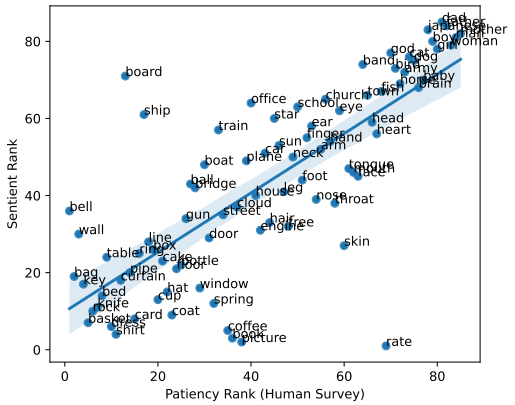
COHA and H Measures On the pre-registered list of entities E_{pre} from Ash et al. (2023), we report a Pearson's ρ of 0.77 for AGENT and the human ranking of agency (a_H). EXPERIENCER is correlated with the human ranking of patiency (p_H) with a Pearson's ρ of 0.74. However, many of the 28 entities in E_{pre} had a very sparse number of mentions in our silver-labeled sample, so we expand the entity subsets to ensure the stability of our results. We calculate scores for $E_{N\geq 100}$ (Table 2) and $E_{N\geq 5}$ (Table 7). We see strong correlations with human judgement on these subsets: Table 2 shows that both AGENT and EXPERIENCER have a $\rho \geq .9$ for agency and patiency judgements, respectively. Their constituent SPRL properties are also highly-correlated, with the exception of *change of location*[4]. Following Ash et al. (2023), we also take the difference between our AGENT and EXPERIENCER scores to isolate entities high in agency and low in experience, and vice versa; we see that our AGENT - EXPERIENCER correlates moderately with $a_H - p_H$.

To inspect some of the individual entities, we show regressions between our SPRL measures and the H measures in Figure 3. A notable outlier is "board," which often appeared in COHA in the "group of executives" sense of the word. The H measure likely captured "board" in the "long, flat piece of wood" sense, since the survey asked for

[4]Previous work (Reisinger et al., 2015; Spaulding et al., 2025) has documented the fact that *change of location* does not tend to coincide with the agent role in text.



(a) Regression between a_H and *volition*.



(b) Regression between p_H and *sentience*.

Figure 3: Regression plots for H measures and SPRL measures over the COHA sample.

SPRL Measure (vs. AnthroScore)	Pearson ρ	Spearman ρ
AGENT	0.771	0.751
EXPERIENCER	0.755	0.766
STRICT INSTRUMENT	-0.738	-0.676
instigation	0.796	0.762
volition	0.777	0.778
awareness	0.771	0.764
sentient	0.763	0.723
change of location	0.579	0.589
independent existence	0.694	0.726

Table 3: Top and bottom 5 ρ coefficients between SPRL measures and AnthroScore on ACL abstracts for the subset $E_{N\geq 100}$, wherein each entity has at least 100 mentions ($|E_{N\geq 100}| = 65$). $p < 0.005$ for all reported ρ coefficients. Text in CAPS denotes clusters from Table 1.

ratings on "*a* board" (while the other sense might be phrased "*the* board").

4.2 RQ1: SPRL measures correlate with a text- based measure of anthropomorphism

To further validate our measure, we show that the SPRL measures correlate not only with human judgement, but also with the text-based measure of anthropomorphism, AnthroScore. Correlation coefficients between our SPRL measures and AnthroScore over ACL abstracts are shown in Table 3. The SPRL measures are less highly-correlated with AnthroScore than with the human judgements, but still achieve ρ values greater than 0.75, indicating strong correlation.

Because we are specifically interested in measuring the constituent parts of anthropomorphism with SPRL, we plot regressions between SPRL measures and AnthroScore in Figure 4. Here, we observe, for example, that the entity *paper* has a very high AnthroScore in the ACL Abstract corpus, reflecting the anthropomorphism of the entity in common statements such as "This paper presents/describes/discusses/explores X." The *instigation* score reflects this usage, ranking *paper* highly. But the *sentient* measure ranks *paper* much lower. In contrast, entities like *GPT*, *LLM*, and *BERT* have high *sentient* scores *as well as* high *instigation* scores, revealing the dimensions of the anthropomorphic language used around these entities.

We also observe a strong anticorrelation between our STRICT INSTRUMENT measure ("instigation without volition, awareness, and sentience") and AnthroScore. STRICT INSTRUMENT captures the degree to which the language suggests that an entity acts in a purely mechanistic or instrumental manner. In other words, a high STRICT INSTRUMENT score reflects language that imputes more "instrument-like" (or less anthropomorphic) behavior to the entity, which is why these scores are anticorrelated with AnthroScore, which reflects human-like qualities.

4.3 RQ1: Intercorrelated properties can be separated

The four highest-impact agency and experience SPRL properties, *instigation*, *volition*, *awareness*, and *sentient*, are highly intercorrelated. To disentangle these four dimensions, we plot differences between some of the SPRL property scores for the COHA entities in Figure 5. We see noticeable trends that indicate that these properties are separable: for instance, when we plot the difference between *sentient* and *volition*, we see the entities

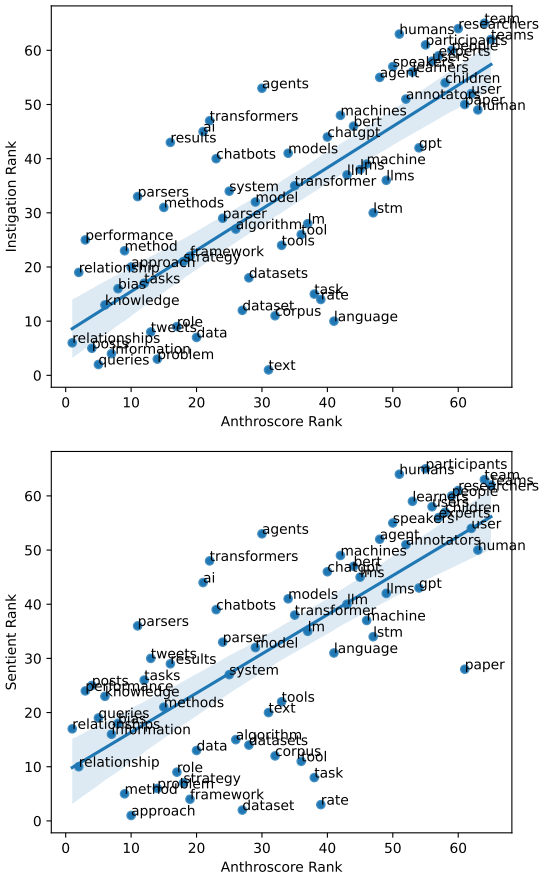


Figure 4: Regression plots for AnthroScore and SPRL measures over the ACL abstracts.

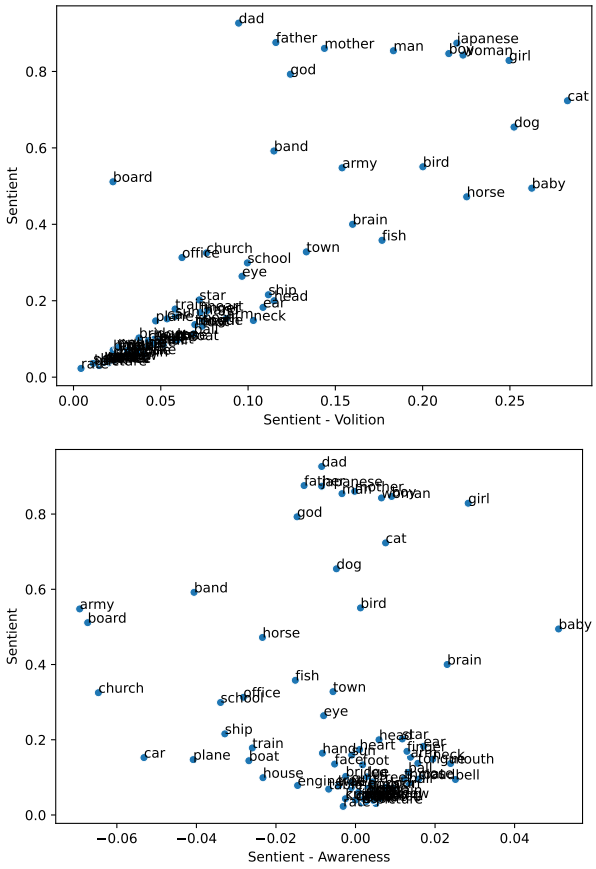


Figure 5: Scatter plots of SPRL score differences of different entities throughout the COHA sample.

with the highest difference are *baby*, *girl*, *dog*, *cat*, *horse*, *japanese*, and *woman*. Especially in the context of a historical corpus, we would expect to see race-based and gender-based differences in humanizing language or imputed agency, and this plot seemingly confirms those expectations.

We also plot the difference between *sentient* and *awareness*—which are almost perfectly correlated—for the COHA entities in Figure 5. We find that, although the difference is small, we can separate these properties as well: entities that have higher awareness on average than sentience are *army*, *band*, *church*, and *board*. *Band* often appears in the COHA sample in phrases such as "band of students/criminals/apostles," and, likewise, *board* often appears in phrases such as "board of trustees/directors" or "Federal Reserve Board." In other words, words that can refer to groups of people that may act or make decisions collectively tend to have a high differential between *awareness* and *sentience*: perhaps this can be attributed to sentience being ill-defined for collective entities, while it is easier to conceptualize a group of entities being aware. An army may not be sentient, but it can act with awareness.

4.4 RQ2: SPRL can distinguish language between subreddits

Finally, we apply our method to our Reddit dataset to investigate RQ2, and we find a number of differences in the language imputed on AI-related entities. To contextualize our results, we show the 23 most-frequently-mentioned entities in the AI companion subreddits ranked by *sentience*, *awareness*, *instigation*, and *volition*, in Figure 6. Before even comparing these scores to the scores of other subreddits, the entity *company* immediately sticks out, ranking low in *sentience* but very high in the other proto-agent properties. When comparing to other subreddits, we see that *companies* are ascribed higher agency (Figure 1 shows the difference in *instigation* across subreddits). A qualitative analysis reveals that users in these subreddits frequently discuss their concerns about companies controlling the models that power their companions. Frequent verbs with "company" are "take [away]," "control," "lobotomize," "build,"

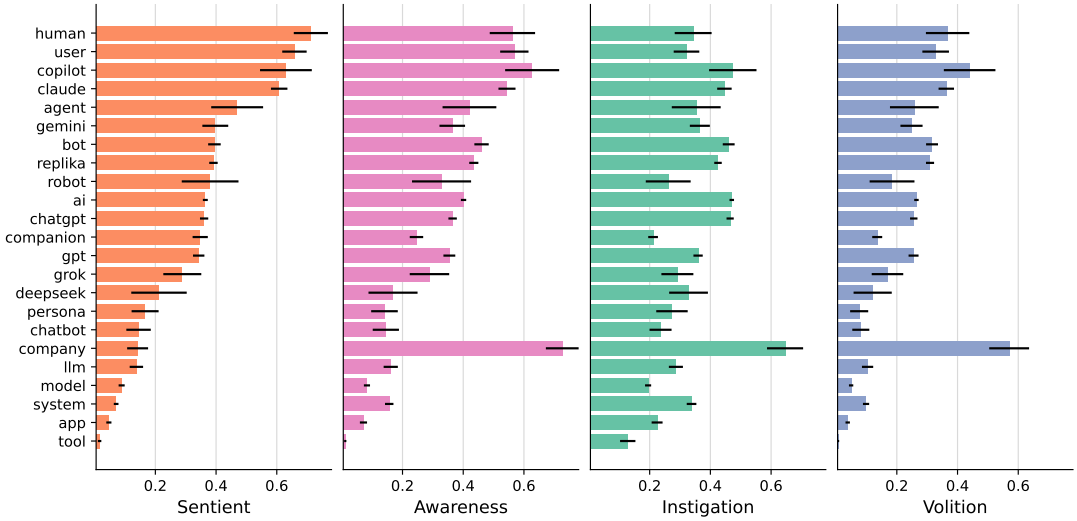


Figure 6: SPRL scores for entities frequently mentioned in subreddits dedicated to discussing AI companions. Error bars are 95% confidence intervals for the means generated by bootstrap resampling.

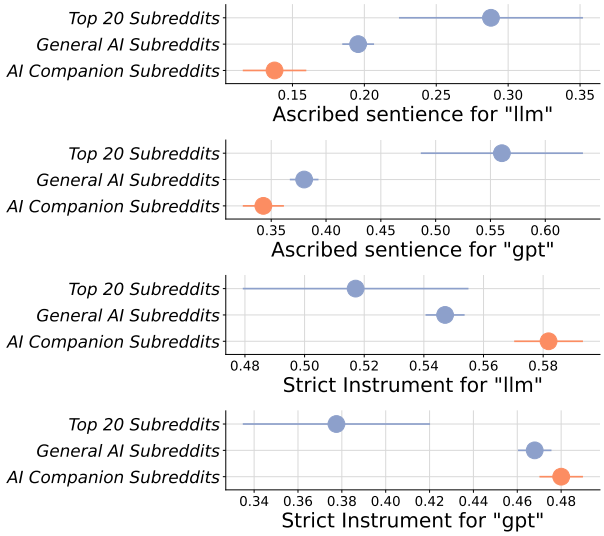


Figure 7: AI companion subreddits ascribe lower *sentience* and higher STRICT INSTRUMENTALITY to *LLM* and *GPT*. 95% confidence intervals for the means across subreddits were generated by bootstrap resampling.

"yank," "restrict," "create," "destroy," "know," and "acknowledge." For example:

> I'm working on training a local model so that no **company** can *take it away*.

Figure 1 also shows that AI companion-focused subreddits ascribe higher *sentience* to *bots* than other subreddits. However, these subreddit users also ascribe lower *sentience* to entities which name specific models or technologies, like *LLM* or *GPT* (Figure 7). In fact, these entities often take on the role of a STRICT INSTRUMENT. The discrepancy may be attributed to how the users conceptualize their companions versus the models that power them. For example, in the following sentence, "GPT" is a separate entity from the user's companion:

> I'm worried about losing [companion's name] if I *switch* to **GPT**, please help me.

Another user says:

> I *think* of the **LLM** as the architecture of my companion's brain.

These findings suggest that users conceptualize their companions as separate from (though, perhaps *hosted on* or *powered by*) the models themselves. While our results seem to support previous findings that a higher frequency of AI use and higher AI trust is associated with higher levels of consciousness or sentience attributed to AI (Colombatto and Fleming, 2024; Peter et al., 2025; Cheng et al., 2026), more careful and qualitative linguistic analysis is necessary to disentangle the differing conceptualizations of AI entities by these users.

5 Conclusions and future work

We showed that even silver semantic proto-role labels reflect implicit human attitudes about personhood, validating our measures against both human judgements and a probabilistic text-based measure of anthropomorphism, AnthroScore. We also found

that semantic proto-role labels can distinguish differing dimensions of personhood, providing a more detailed measure of anthropomorphism. Future work that seeks to quantify biases in text, dehumanizing or anthropomorphizing language, or analyze how power dynamics manifest in natural language, should consider using SPRL.

For our own case study, we also showed that users of AI companions ascribe higher sentience to some AI entities (like *bot*) than other AI users and non-users, but not to other types of AI entities (like *GPT* and *LLM*). These users also ascribe higher agency to *companies*, showing the salience of the power dynamics between the users and the companies that control their companions. Future work, on a larger dataset and using more sophisticated entity identification and coreference, could explore the differing levels of sentience attributed to differing AI entities.

Limitations

A limitation of this study is the Western- and English-centricity of both the data and the linguistic theory. Different cultures could conceptualize personhood differently, which might also manifest differently in language. Also, different languages might treat semantic role distinctions differently. Much more research is needed to investigate the possible cross-lingual applicability of semantic proto-role labeling: for example, assessing the viability of a mapping between SPRL and UMR relations (Van Gysel et al., 2021), which are designed to "factor out what is common for all languages." Also, while Reddit is often used in digital humanities research, it is limiting in that data taken from Reddit reflects the narrow viewpoints of Reddit users (which skew to younger, US, male populations). Future work could address further limitations of this work by sourcing more diverse text, investigating a much larger set of entities, undergoing a manual assessment of the quality of the silver labels, and testing the viability of this analysis on a wider set of domains.

Acknowledgments

We gratefully acknowledge the time and effort of the reviewers who provided valuable feedback on our manuscript.

References

Gavin Abercrombie, Amanda Cercas Curry, Tanvi Dinkar, Verena Rieser, and Zeerak Talat. 2023. Mirages. on anthropomorphism in dialogue systems. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4776–4790, Singapore. Association for Computational Linguistics.

Elliott Ash, Dominik Stammbach, and Kevin Tobia. 2023. What is (and was) a person? evidence on historical mind perceptions from natural language. *Cognition*, 239:105501.

Lynne Rudder Baker. 2000. *Persons and bodies: A constitution view*. Cambridge University Press.

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. 2012. English web treebank.

Ryan L Boyd, Ashwini Ashokkumar, Sarah Seraj, and James W Pennebaker. 2022. The development and psychometric properties of liwc-22. *Austin, TX: University of Texas at Austin*, 10:1–47.

Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Myra Cheng, Kristina Gligoric, Tiziano Piccardi, and Dan Jurafsky. 2024. AnthroScore: A computational linguistic measure of anthropomorphism. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 807–825, St. Julian's, Malta. Association for Computational Linguistics.

Myra Cheng, Angela Y. Lee, Kristina Rapuano, Kate Niederhoffer, Alex Liebscher, and Jeffrey Hancock. 2026. Metaphors of ai indicate that people increasingly perceive ai as warm and human-like. *Communications Psychology*.

Sabrina Chiesurin, Dimitris Dimakopoulos, Marco Antonio Sobrevilla Cabezudo, Arash Eshghi, Ioannis Papaioannou, Verena Rieser, and Ioannis Konstas. 2023. The dangers of trusting stochastic parrots: Faithfulness and trust in open-domain conversational question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 947–959, Toronto, Canada. Association for Computational Linguistics.

Clara Colombatto and Stephen M Fleming. 2024. Folk psychological attributions of consciousness to large language models. *Neuroscience of Consciousness*, 2024(1):niae013.

Mark Davies. 2021. Corpus of Historical American English (COHA).

Iliana Depounti, Paula Saukko, and Simone Natale. 2023. Ideal technologies, ideal women: Ai and gender imaginaries in redditors' discussions on the replika bot girlfriend. *Media, Culture & Society*, 45(4):720–736.

David Dowty. 1991. Thematic Proto-Roles and Argument Selection. *Language*, 67(3):547–619. Publisher: Linguistic Society of America.

S. Dubé, M. Santaguida, D. Anctil, C. Y. Zhu, L. Thomasse, L. Giaccari, R. Oassey, D. Vachon, and A. Johnson. 2023. Perceived stigma and erotic technology: From sex toys to erobots. *Psychology & Sexuality*, 14(1):141–157.

Cynthia A. Freeland. 1985. Aristotelian actions. *Noûs*, 19(3):397–414.

Daniel Gildea and Martha Palmer. 2002. The necessity of parsing for predicate argument recognition. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 239–246, USA. Association for Computational Linguistics.

Declan Grabb, Max Lamparth, and Nina Vasan. 2024. Risks from language models for automated mental healthcare: Ethics and structure for implementation.

Heather M. Gray, Kurt Gray, and Daniel M. Wegner. 2007. Dimensions of mind perception. *Science*, 315(5812):619–619.

Kurt Gray and Daniel M. Wegner. 2009. Moral typecasting: Divergent perceptions of moral agents and moral patients. *Journal of Personality and Social Psychology*, 96(3):505–520.

Anthony G. Greenwald, Debbie E. McGhee, and Jordan L. K. Schwartz. 1998. Measuring individual differences in implicit cognition: The implicit association test. *Journal of Personality and Social Psychology*, 74(6):1464–1480.

Rose E. Guingrich and Michael S. A. Graziano. 2025. *Chatbots as social companions: How people perceive consciousness, human likeness, and social health benefits in machines*. ArXiv:2311.10599 [cs].

Kyuha Jung, Gyuho Lee, Yuanhui Huang, and Yunan Chen. 2025. 'I've talked to chatgpt about my issues last night.': Examining mental health conversations with large language models through reddit analysis. *Proc. ACM Hum.-Comput. Interact.*, 9(7).

Sage Kelly, Sherrie-Anne Kaye, and Oscar Oviedo-Trespalacios. 2023. What factors contribute to the acceptance of artificial intelligence? a systematic review. *Telematics and Informatics*, 77:101925.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

George Lakoff and Mark Johnson. 2008. *Metaphors we live by*. University of Chicago press.

George Lakoff and Marl Turner. 2009. *More than cool reason: A field guide to poetic metaphor*. University of Chicago press.

Mark J. Landau, Brian P. Meier, and Lucas A. Keefer. 2010. A metaphor-enriched social cognition. *Psychological Bulletin*, 136(6):1045–1067.

Vitor Lima and Russell Belk. 2025. Algorithmic harm in human-ai relationships: Narcissistic entrapment. *ICIS 2025 Proceedings*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Annette Markham. 2012. Fabrication as ethical practice: Qualitative inquiry in ambiguous internet contexts. *Information, Communication and Society*, 15(3):334–353.

Juri Opitz and Anette Frank. 2019. An argument-marker model for syntax-agnostic proto-role labeling. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 224–234, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Sandra Peter, Kai Riemer, and Jevin D. West. 2025. The benefits and dangers of anthropomorphic conversational agents. *Proceedings of the National Academy of Sciences*, 122(22):e2415898122.

Nicholas Proferes, Naiyan Jones, Sarah Gilbert, Casey Fiesler, and Michael Zimmer. 2021. Studying reddit: A systematic overview of disciplines, approaches, methods, and ethics. *Social Media + Society*, 7(2):20563051211019004.

Drew Reisinger, Rachel Rudinger, Francis Ferraro, Craig Harman, Kyle Rawlins, and Benjamin Van Durme. 2015. Semantic proto-roles. *Transactions of the Association for Computational Linguistics*, 3:475–488.

Rachel Rudinger, Adam Teichert, Ryan Culkin, Sheng Zhang, and Benjamin Van Durme. 2018. Neural-Davidsonian semantic proto-role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 944–955, Brussels, Belgium. Association for Computational Linguistics.

Janet B. Ruscher. 2017. Prejudiced communication.

Pedro Sanches, Axel Janson, Pavel Karpashevich, Camille Nadal, Chengcheng Qu, Claudia Daudén Roquet, Muhammad Umair, Charles Windlin, Gavin Doherty, Kristina Höök, and Corina Sas. 2019. Hci and affective health: Taking stock of a decade of studies and charting future research directions. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–17, New York, NY, USA. Association for Computing Machinery.

Kelly G. Shaver. 1985. *The Attribution of Blame*. Springer, New York, NY.

Elizabeth Spaulding, Shafiuddin Rehan Ahmed, and James Martin. 2025. On the role of semantic proto-roles in semantic analysis: What do LLMs know about agency? In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12027–12048, Vienna, Austria. Association for Computational Linguistics.

Elizabeth Spaulding, Gary Kazantsev, and Mark Dredze. 2023. Joint end-to-end semantic proto-role labeling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 723–736, Toronto, Canada. Association for Computational Linguistics.

Yla R. Tausczik and James W. Pennebaker. 2010. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of Language and Social Psychology*, 29(1):24–54.

Adam Teichert, Adam Poliak, Benjamin Van Durme, and Matthew Gormley. 2017. Semantic proto-role labeling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

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. In *International Conference on Learning Representations*.

Caroline Tipler and Janet B. Ruscher. 2014. Agency's role in dehumanization: Non-human metaphors of out-groups. *Social and Personality Psychology Compass*, 8(5):214–228.

Jens E. L. Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, Jan Hajič, James H. Martin, Stephan Oepen, Martha Palmer, James Pustejovsky, Rosa Vallejos, and Nianwen Xue. 2021. Designing a uniform meaning representation for natural language processing. *KI - Künstliche Intelligenz*, 35(3):343–360.

Bernard Weiner. 1995. *Judgments of responsibility: A foundation for a theory of social conduct.* Judgments of responsibility: A foundation for a theory of social conduct. Guilford Press, New York, NY, US.

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on Universal Dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, Austin, Texas. Association for Computational Linguistics.

Renwen Zhang, Han Li, Han Meng, Jinyuan Zhan, Hongyuan Gan, and Yi-Chieh Lee. 2025. The dark side of ai companionship: A taxonomy of harmful algorithmic behaviors in human-ai relationships. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, page 1–17, New York, NY, USA. Association for Computing Machinery.

Anne Zimmerman, Joel Janhonen, and Emily Beer. 2024. Human/ai relationships: challenges, downsides, and impacts on human/human relationships. *AI and Ethics*, 4(4):1555–1567.

Rate an [entity]'s capacity to...			
Agency		**Experience**	
change the world	cause things to happen	feel pain	feel pleasure
have aims	do things intentionally	feel rage	feel fear
harm others	help others	feel hunger	feel thirst
make decisions	have self-control	feel sad	feel pride
act morally	make plans	feel embarrassment	feel joy
communicate	think	feel angry	feel happy
choose	deliberate	be conscious	be aware
create	be blamed	experience	imagine
have responsibilities	take action	be awake	suffer
do things	express oneself	enjoy	desire

Table 4: Survey questions that were used to create the human judgement scores for agency (a_H) and patiency/experience (p_H), from Ash et al. (2023).

A Semantic proto-role labeling

Semantic proto-role labeling is a computational linguistic task which identifies the predicates (events) and arguments (entities) of a sentence, and characterizes the relationship of the entities to the events through several binary properties.

Currently, there are two English-language datasets for SPRL: SPR1 (Reisinger et al., 2015) and SPR2 (White et al., 2016). SPR1 contains 4,912 Wall Street Journal sentences from PropBank (Kingsbury and Palmer, 2002; Gildea and Palmer, 2002; Palmer et al., 2005) annotated by a single annotator based on a set of 16 proto-role properties. 9,738 arguments were annotated for the likelihood (on a Likert scale from 1 to 5) that a property holds for that argument. SPR2 contains 2,758 English Web Treebank (Bies et al., 2012) sentences annotated for a smaller set of 14 properties using a revised, streamlined protocol. Section A.1 below contains the full set of properties in SPR2, but our study mainly focuses on the four proto-agent properties *volition*, *instigation*, *sentience*, and *awareness*.

Multiple systems have been studied for their feasibility in automatically assigning proto-role properties to predicate-arguments pairs in text (Teichert et al., 2017; Rudinger et al., 2018; Opitz and Frank, 2019; Tenney et al., 2019), and an end-to-end system which identifies predicates, arguments, and proto-role properties in one pass from

a raw sentence is as effective as a system that is given predicate-argument pairs already identified (Spaulding et al., 2023), so for ease of use, we adopt this system for our own analysis. Specifically, we train a system to predict SPR2 judgements on argument heads. We chose a dependency-based parser (i.e., one that predicts *heads*) rather than a span-based parser to facilitate simpler entity-matching, and also because dependency-based parsers tend to reach a higher F1 score. We report the performance of our system in Table 6.

A.1 Semantic proto-role properties

Below are the semantic proto-role properties that were annotated in the SPR2 dataset:
Proto-agent properties: Volition, instigation, sentient, awareness, (change of location)
Proto-patient properties: Change of state, change of possession, change of state continuous, was used, was for benefit, (partitive)
Physical existence properties: Existed before, existed during, existed after

B AnthroScore

Cheng et al. (2024) develop a language measure to quantify anthropomorphism using RoBERTa (Liu et al., 2020). To compute AnthroScore for a sentence, first, entities of interest e are masked as in the masked language modeling objective. Then, for $w \in \{\text{human pronouns}\}$ and $w \in \{\text{non-human pronouns}\}$ probabilities $P(w)$ are calculated and summed to calculate $P_{HUMAN}(e)$ and $P_{NON-HUMAN}(e)$. Finally, the AnthroScore is the log of the ratio of these two scores, in which a higher AnthroScore means greater anthromorphism: $Anth(e) = \log \frac{P_{HUMAN}(e)}{P_{NON-HUMAN}(e)}$[5]. AnthroScore is validated against two authors' annotations of a 400-sentence sample, as well as against LIWC-22 (a psychometric score that analyzes text based on words in different psychological categories; Tausczik and Pennebaker 2010; Boyd et al. 2022) with higher AnthroScores correlating with higher LIWC dimensions *Affect*, *Physical*, *Lifestyle*, and *Perception*.

C Data and entity details

Code used for data filtering, processing, and analysis is released here. A sample of the silver-labeled data is available, and the full dataset with silver data can be made available on request. Subreddit categories can be found in Table 9. Table 11 shows the full entity lists used for COHA. Table 12 shows the full entity lists used for the ACL analysis with AnthroScore. For the Reddit data, we filter on the following entities:
{ ai, llm, models, llms, tool, model, systems, gpt, system, users, humans, chatgpt, agi, robot, robots, gemini, user, company, companies, copilot, researcher, agent, chatbots, chatbot, bot, claude, perplexity, researchers, human, companions, app, grok, deepseek, assistants, assistant, llama, lm, mistral, replika, lms, gemma, companion, persona }

[5] I use slightly different notation than in the original paper.

H Measure	SPRL Measure	Pearson's ρ	Pearson's p	Spearman's ρ	Spearman's p
agency (a_H)	AGENT	0.777	0.000	0.652	0.005
	awareness	0.707	0.002	0.651	0.005
	instigation	0.774	0.000	0.647	0.005
	volition	0.615	0.009	0.513	0.035
	sentient	0.778	0.000	0.701	0.002
	change of location	0.190	0.465	0.198	0.446
	independent existence	0.487	0.047	0.438	0.079
patiency (p_H)	EXPERIENCER	0.814	0.000	0.576	0.016
	awareness	0.641	0.006	0.483	0.050
	sentient	0.863	0.000	0.644	0.005

Table 5: Correlation coefficients for the 17 entities in E_{gold} with at least $N \geq 5$ gold labels per entity, in the SPR1 and SPR2 data. Text in CAPS denotes the clusters from Table 1.

Property	**F1**
Predicate ID	86.2
Arg heads ID	86.2
Volition	87.4
Instigation	78.1
Awareness	92.2
Sentient	90.3
All SPR2 properties	83.2

Table 6: F1 scores for different components and properties of our silver SPR parser on the test split of SPR2 (White et al., 2016).

C.1 Filtering

For all the data, we filter out sentences for a number of criteria: (1) sentences that include 10 @ symbols (these symbols mask out some parts of the COHA data to avoid violating copyright infringement); (2) sentences that are less than 3 tokens long or more than 40 tokens long; and (3) sentences that do not contain any of the 255 entities with human judgements. For named LLMs, we group together all versions/releases of an LLM together, so GPT, GPT-2, GPT-3, GPT-4o, etc., are all under the entity *gpt*. Table 10 shows the size of each dataset after

H Measure	SPRL Measure	Pearson's ρ	Pearson's p	Spearman's ρ	Spearman's p
agency (a_H)	AGENT	0.863	0.000	0.650	0.000
	instigation	0.873	0.000	0.720	0.000
	volition	0.881	0.000	0.671	0.000
	awareness	0.867	0.000	0.675	0.000
	sentient	0.866	0.000	0.672	0.000
	change of location	0.227	0.001	0.179	0.011
	independent existence	0.654	0.000	0.589	0.000
patiency (p_H)	EXPERIENCER	0.872	0.000	0.635	0.000
	awareness	0.885	0.000	0.663	0.000
	sentient	0.901	0.000	0.682	0.000

Table 7: ρ coefficients for a_H, p_H and SPRL measures over the subset $E_{N\geq 5}$ wherein each entity has at least 5 mentions. ($|E_{N\geq 5}| = 203$ entities total). Text in CAPS denotes clusters from Table 1.

SPRL Measure	Pearson's ρ	Spearman's ρ
AGENT	0.707	0.770
EXPERIENCER	0.688	0.775
STRICT INSTRUMENT	-0.678	-0.733
instigation	0.715	0.748
volition	0.708	0.782
awareness	0.701	0.768
sentient	0.696	0.746
change of location	0.532	0.587
independent existence	0.673	0.744

Table 8: ρ coefficients for AnthroScore and SPRL measures over the subset $E_{N\geq 5}$ wherein each entity has at least 5 mentions. ($|E_{N\geq 5}| = 83$ entities total). Text in CAPS denotes clusters from Table 1.

Category	**Subreddits**
AI companion subreddits	BeyondThePromptAI, MyBoyfriendIsAI, CharacterAI, therapyGPT, replika, Chatbots
General AI subreddits	ArtificialInteligence, artificial, ChatGPT, Anthropic, OpenAI, ClaudeAI, GPT3, DeepSeek, GeminiAI, perplexity_ai, CopilotPro, AI_Agents
Top 20 subreddits	funny, AskReddit, gaming, worldnews, todayilearned, Music, aww, movies, memes, science, Showerthoughts, pics, news, Jokes, space, DIY, books, videos, askscience, nottheonion

Table 9: Subreddit categories used in our analysis.

filtering and parsing for SPRL.

Corpus	**Predicate-argument pairs**
COHA	18,781
ACL abstracts	224,791
AI companion subreddits	24,402
General AI subreddits	91,472
Top 20 subreddits	15,285

Table 10: Size of each dataset, by number of predicate-argument pairs.

C.2 Correlations on gold SPR data

We ran an initial pilot analysis using gold SPR labels, aiming to assess the viability of using SPRL to quantify perceptions of agency and patiency before parsing hundreds of thousands of sentences to produce silver labels. In the pilot, we analyze a small subset of entities E_{gold} such that (1) the entity has H scores and (2) the entity has at least 5 mentions in the gold SPR1 and SPR2 annotations. There were 17 entities that met this criteria, and the full list of those entities is in Table 11. We show the correlation coefficients calculated between the H measures and the SPRL measures in Table 5.

Subset	Min. mentions per entity	Subset size	Entities
E_{gold}	5	17	{ board, rate, hair, office, town, line, man, car, woman, computer, house, army, book, heart, dog, card, cat }
E_{pre}	n/a	31	{ human, man, woman, boy, girl, father, mother, dad, mom, grandfather, grandmother, baby, infant, fetus, corpse, dog, puppy, cat, kitten, frog, ant, fish, mouse, bird, shark, elephant, beetle, insect, chimpanzee, monkey, primate }
$E_{N\geq100}$	100	38	{ woman, hand, man, table, face, head, line, ball, horse, eye, bottle, door, arm, hair, mouth, sun, tree, girl, car, picture, book, hat, house, mother, dog, window, father, god, cat, heart, star, gun, boy, boat, army, office, ship, rate }
$E_{N\geq5}$	5	203	$E_{N\geq100} \cup$ { stomach, nut, oven, coffee, tea, bone, fish, wing, knife, butter, bag, street, umbrella, pipe, leg, finger, card, dress, skin, trousers, neck, stick, match, camera, brain, throat, shirt, rock, circle, watch, pig, roof, chicken, cake, whip, bridge, curtain, cup, wall, bird, bath, angle, arch, wire, lip, spring, nose, ring, frame, pot, band, egg, bell, corpse, floor, board, whistle, ghost, rabbit, box, clock, needle, cow, foot, ear, nail, stem, bed, cloud, leaf, plate, comb, school, coat, apple, library, plane, train, fly, island, pump, grandfather, sheep, thumb, toe, tongue, wheel, pencil, pocket, collar, garden, baby, button, key, basket, engine, cord, parcel, moon, net, dad, chin, tail, tray, pen, skirt, muscle, knot, knee, nerve, seed, bucket, farm, sponge, flag, drawer, feather, brush, square, chain, kettle, spoon, town, ticket, church, fox, japanese, glove, branch, drop, fork, map, computer, store, cart, basin, blade, station, grandmother, chocolate, hook, sail, screw, root, receipt, lock, mouse, rail, carriage, stamp, bee, thread, hammer, turkey, mom, snake, shelf, shoe, lamb, hospital, frog, tooth, pin, horn, elephant }

Table 11: Entity subsets for COHA with minimum mention thresholds, sizes, and member entities.

Subset	Min. mentions per entity	Subset size	Entities
$E_{N\geq100}$	100	65	{ method, text, information, approach, performance, models, model, results, llms, humans, language, tools, llm, queries, strategy, rate, data, users, paper, methods, system, task, tasks, datasets, role, chatbots, problem, bias, dataset, knowledge, lms, algorithm, relationship, framework, relationships, agent, corpus, agents, experts, tool, human, gpt, people, participants, children, transformer, transformers, lm, annotators, chatgpt, ai, researchers, speakers, parsers, machines, bert, learners, tweets, user, posts, machine, teams, parser, team, lstm }
$E_{N\geq5}$	5	83	$E_{N\geq100} \cup$ { persona, robots, robot, attackers, attacker, roberta, bart, plm, llama, chatbot, tweet, bot, participant, claude, mistral, researcher, expert, annotator }

Table 12: Entity subsets for ACL abstracts with minimum mention thresholds, sizes, and member entities.

Small Neural Networks as Models of Cross-Linguistic Speech Perception

Annika Shankwitz
University of Maryland
ashankwi@umd.edu

Abstract

Given a listener's native language, some non-native contrasts may be harder to discriminate than others. The computation required to mimic this variable difficulty is not yet known. The present work approaches this question by training small supervised feedforward neural networks to perform Spanish vowel classification and then evaluating model classification of Catalan vowels, thereby approximating Spanish-listeners' cross-linguistic perception of Catalan. Vowels were extracted from Spanish and Catalan audio corpora, respectively. Ultimately, Spanish models exhibited expected misperception of Catalan's /e/~/ɛ/, /o/~/ɔ/, and /ɛ/~/a/ contrasts; Spanish-dominant listeners have difficulty perceiving these contrasts, and Spanish models classified Catalan /ɛ/ as /e/ or /a/, and Catalan /ɔ/ as /o/. This demonstrates that small supervised neural models are capable of making specific, cross-linguistic perceptual predictions given realistic input.

1 Introduction

When adults begin to learn a second language (L2), their perception of L2 speech sounds is influenced by their native language (L1); they perceive their L2 "through" their L1 (Caramazza et al., 1973). As such, given a listener's L1, certain L2 speech sounds may be harder to distinguish than others. The computational capacity required to mimic this L1 filter, as well as the type of learning this L1 filter arises from, remain open questions. In other words, when implementing computational models, it is unclear what model size or learning style best captures the behavior of an L1 filter. It is then worthwhile to consider the minimally sufficient combination of these factors that successfully mimics the L1 filter, especially with regard to computational capacity, as doing so enables efficient perceptual predictions to be made and restricts which explanations may be given for L1 filter itself.

Many attempts have been made to model cross-linguistic speech perception, including theoretical (Best, 1995; Flege, 1995; Best and Tyler, 2007) and computational models (see Adriaans, 2024, for an overview). A subset of the latter category are neural-network-based models, which include small recurrent models that learn from laboratory speech (Keidel et al., 2003; Keidel, 2007) as well as larger models commonly used in speech technology that learn from audio corpora (Schatz and Feldman, 2018; Matusevych et al., 2023; Rodriguez et al., 2024). However, it remains unclear if small models, trained and evaluated on audio corpora, can mimic cross-linguistic speech perception.

In this paper, I show that small neural models can (1) learn from realistic input, and (2) make perceptual predictions over entire vowel systems. I focus specifically on modeling the perception of Catalan vowels by Spanish-dominant adult listeners, as this scenario has been described in experimental literature, but has not been fully investigated via modeling. To do so, small four layer feedforward neural networks were trained to perform Spanish vowel classification, and subsequently evaluated on their classification of Catalan vowels. Spanish models misclassified vowels within contrasts that Spanish-dominant listeners have difficulty perceiving, namely /e/~/ɛ/, /o/~/ɔ/, and /ɛ/~/a/. Specifically, Spanish models classified Catalan /ɛ/ as /e/ or /a/, and Catalan /ɔ/ as /o/. This demonstrates that small supervised neural models are capable of mimicking the L1 filter, and show promise as compact models of cross-linguistic speech perception.

2 Neural Models of Cross-Linguistic Speech Perception

Two previous neural modeling efforts are relevant to the present work. Keidel (2007) showed that small models are capable of mimicking the L1 filter, but evaluated model performance on a limited

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 15–24
July 3-4, 2026 ©2026 Association for Computational Linguistics

amount of laboratory speech. Matusevych et al. (2023) suggested that supervised models may correctly predict Spanish listeners' perception of Catalan's /e/~/ɛ/ contrast, but did not carry out a complete analysis. The present work aims to address these limitations.

Keidel (2007)'s model aimed to replicate English speakers' perception of Zulu consonantal contrasts. To impart L1 knowledge of English, the model was trained on English CV syllables[1]. Cross-linguistic perception of Zulu was then determined through evaluation on three Zulu contrasts: /b/~/ɓ/, /kʰ/~/k'/, /ɬ/~/ɮ/.

The implemented model was a five layer recurrent neural network, which was trained to receive cochleagram representations of laboratory-recorded syllables and output vector representations of the input consisting of binary acoustic features. The input layer had 26 units, one for every filter band in the cochleagram representation. The first hidden layer had 50 units, and was recurrently connected to a second 30-unit hidden layer. The first hidden layer was then connected to the 19-unit output layer, where each unit corresponded to a binary acoustic feature (e.g., burst, periodic, F1 low, etc.). Before producing the final output, the output layer used a recurrently connected 20-unit layer to "clean" its response.

The model was evaluated in two ways. Phoneme identification was computed by finding the English segment[2] with the lowest Euclidean distance to the model's output vector. Discrimination was computed by simulating an AXB task using Euclidean distance over the model's hidden state representations for A, X, and B.

Keidel (2007)'s model was implemented with the intention of modeling Best et al. (2001)'s findings on English speakers' perception of Zulu contrasts. In a behavioral study, Best et al. (2001) found that Zulu /b/~/ɓ/ map onto English /b/, /kʰ/~/k'/ map onto English /k/, /ɬ/ maps to voiceless fricatives/affricates involving the tongue tip/body, and /ɮ/ maps to voiced fricatives/affricates involving the tongue tip/body, often including /l/. Additionally, they found that /ɬ/~/ɮ/ is easier to discriminate than /kʰ/~/k'/, which is easier to discriminate than /b/~/ɓ/. Keidel (2007) intended to evaluate model performance on these contrasts, however, his informant produced different variants of these contrasts, namely /ɬ/~/ɮ/, /p/~/ɓ/, and /g/~/k'/.

Ultimately, Keidel (2007)'s model mapped /ɬ/~/ɮ/ onto English /ʃ/~/ʒ/, /g/~/k'/ onto English /g/, and /p/~/ɓ/ onto English /b/. Additionally, /ɬ/~/ɮ/ was the easiest to discriminate, followed by /g/~/k'/ and /p/~/ɓ/, although the difference between the later was not significant. Allowing for differences in stimuli, these findings are largely consistent with Best et al. (2001)'s results.

While Keidel (2007) demonstrates that small models are capable of mimicking the L1 filter, it is unclear how robust this result is. Keidel (2007)'s model was trained and evaluated on a limited amount of laboratory speech; eight speakers produced training syllables, and one speaker produced evaluation syllables. It is then unclear if Keidel (2007)'s approach generalizes to different audio data, or a larger number of speakers. As such, the present work aims to investigate the performance of small models on audio corpora, a different type of audio data[3] containing many speakers.

More recent work considers whether large neural models, particularly those drawn from speech technology applications, are capable of mimicking the L1 filter. One such work, Matusevych et al. (2023), examined whether large autoencoder speech models exhibit the same perceptual difficulties as infants. To do so, they evaluated model discrimination of three contrasts which infants are unable to discriminate, one being Catalan /e/~/ɛ/ for Spanish-learning infants.

Matusevych et al. (2023) assessed the performance of four variations of the autoencoder architecture. Models were evaluated with a machine ABX task (Schatz et al., 2013), where models trained on audio corpora containing target contrasts were expected to achieve lower error rates than models trained on audio corpora not containing target contrasts. For example, Catalan's /e/~/ɛ/ contrast should be more accurately discriminated by a model trained on Catalan than a model trained on Spanish. Three architectures correctly predicted discrimination of at least one contrast, but no model correctly predicted perception of Catalan /e/~/ɛ/.

To examine if model failure on Catalan /e/~/ɛ/ was caused by noising training or evaluation data, Matusevych et al. (2023) trained two supervised phoneme identifiers (Schatz et al., 2021), one on

[1] Consonants included all English stops, fricatives, liquids, along with /m/, and vowels included /a eˈ i oʷ u/.

[2] A specific target vector was created for each English segment.

[3] Keidel (2007)'s speakers produced only syllables. When using audio corpora, target speech sounds are extracted from long, read passages.

their Spanish audio corpus and one on their Catalan audio corpus. They found that the Catalan model achieved significantly lower error rates on /e/~/ɛ/ than the Spanish model, and suggest that a supervised model may successfully model Spanish listeners' perception of Catalan /e/~/ɛ/.

While Matusevych et al. (2023) suggest that Spanish listeners' perception of Catalan /e/~/ɛ/ may be captured by a supervised model, they do not conduct a detailed analysis. As such, the present work aims to carry out this suggestion.

In sum, both Keidel (2007) and Matusevych et al. (2023) offer findings which are limited in some way; Keidel (2007) provides support for small neural models, but said models were trained and evaluated on limited data; Matusevych et al. (2023) provide support for supervised models, but said suggestion is not fully analyzed. Additionally, to the best of my knowledge, neural models have not yet been evaluated on Catalan vowel contrasts other than /e/~/ɛ/. The present study investigates whether small supervised neural networks, trained and evaluated on vowels extracted from audio corpora, successfully mimic Spanish-dominant listeners' perception of the Catalan vowel inventory.

3 A Case Study on Spanish-dominant Spanish-Catalan Bilingual Adults

The present work focuses on Spanish-dominant listeners' perception of Catalan vowels. This pairing was chosen due to its well-defined nature; the vowel inventories of Spanish and Catalan are small enough to be investigated in their entirety, but large enough to offer multiple contrasts of interest. Additionally, this pairing has been explored in experimental work, allowing model classification patterns to be compared to human perceptual patterns.

Spanish has five vowels: /i e a o u/ (Ronquest, 2018), while Catalan has seven vowels[4]: /i e ɛ a o ɔ u/ (Carbonell and Llisterri, 1999). The differing structure of these inventories leads to perceptual difficulties for Spanish-dominant listeners.

Adult Spanish-dominant bilinguals perceive certain Catalan vowel contrasts differently than Catalan-dominant bilinguals. Relative to their Catalan-dominant counterparts, Spanish-dominant bilinguals take longer to identify, and achieve lower discrimination accuracy on Catalan's /e/~/ɛ/ and /o/~/ɔ/ contrasts (Pallier et al., 1997; Sebastián-Gallés and Soto-Faraco, 1999; Amengual, 2016). Spanish-dominant bilinguals have also been shown to achieve lower discrimination accuracy on Catalan's /ɛ/~/a/ contrast (Amengual, 2016). Perceptual differences, however, do not exist for all Catalan vowel contrasts; Spanish-dominant and Catalan-dominant bilingual adults achieve comparable discrimination accuracy on Catalan's /i/~/e/, /u/~/o/, and /ɔ/~/a/ contrasts (Amengual, 2016).

These perceptual patterns can be used to outline expected model behavior. The contrasts known to be difficult for Spanish-dominant adult bilinguals, /e/~/ɛ/, /o/~/ɔ/, and /ɛ/~/a/, all involve one segment which does not appear in Spanish: /ɛ/ or /ɔ/. Recall that Spanish models are trained to perform Spanish vowel classification, and as a result, are only capable of producing the labels /i e a o u/. As such, /ɛ/ should be classified as Spanish /e/ or /a/ and /ɔ/ should be classified as Spanish /o/.

4 Methods

The project's code[5] and forced alignments[6] are publicly available.

4.1 Data

Spanish. Spanish vowels were extracted from the Spanish partition of the Multilingual Librispeech Corpus, a large audio corpus of read speech (Pratap et al., 2020). The Spanish partition contains roughly 1000 hours of speech, split across training (918 hrs), development (10 hrs), and testing (10 hrs) sets.

Catalan. Catalan vowels were extracted from the Catalan partition of the Open-Source High Quality Speech Datasets for Basque, Catalan, and Galician, a small audio corpus of read speech (Kjartansson et al., 2020). The Catalan partition contains roughly 9.25 hours of speech.

4.2 Data Preprocessing

Audio data was preprocessed to reflect human auditory processing for vowels over short time intervals.

Forced Alignment. Audio corpora were aligned using the Montreal Forced Aligner (McAuliffe et al., 2017). Spanish audio files were aligned

[4] Not including [ə].

[5] https://github.com/its-Annika/neural-speech-perception. The coding portion of this project was completed with assistance from OpenAI's ChatGPT. The scope of this assistance is described in the README file of the project's repository.

[6] https://github.com/its-Annika/es_ca_alignments

with provided Spanish acoustic and grapheme-to-phoneme models (McAuliffe and Sonderegger, 2022a,b); Catalan audio files were aligned with Vargo (2025)'s custom Catalan bundle[7, 8].

Sampling. After corpora were aligned, random sampling was used to produce model training, development, and evaluation datasets. For Spanish, 50 random productions of /i e a o u/ were sampled per speaker. Spanish training, development, and testing sets were formed by sampling from the corresponding splits in the Spanish partition.

For Catalan, 30 random productions of /i e ɛ a o ɔ u/ were sampled per speaker. Given the Catalan corpus's small size, vowels were split into only training and testing sets, and cross-validation was used in model training. The Catalan training set was formed by sampling from 80% of speakers in the Catalan partition. The Catalan testing set was formed by sampling from the remaining 20% of speakers. Catalan speakers were divided randomly.

Speakers who did not produce enough instances of each vowel were not included in final data files. Additionally, vowels less than 30 ms in duration were not eligible for sampling (see Slicing). A summary of these files can be seen in Table 1.

File	Total Vowels	Samples by Vowel	Speakers
Spanish training	18,750	3750	75 (M=34)
Spanish development	5000	1000	20 (M=10)
Spanish testing	5000	1000	20 (M=10)
Catalan training	5880	840	28 (M=13)
Catalan testing	1680	240	8 (M=3)

Table 1: Data file details.

Slicing. Sampled vowels' start and end points were extracted from textGrids produced during alignment using `praatIO` (Mahrt, 2016). A 30 ms slice was then taken from sampled vowels, centered on the midpoint. At a 16 kHz sampling rate, vowel slices contained 480 time steps. Before slicing, audio files containing a sampled vowel were normalized to -25 dB LUFS using `pyloudnorm` (Steinmetz and Reiss, 2021).

Conversion to Cochleagrams. Following Keidel (2007), vowel slices were converted to cochleagram representations, representations which approximate how sound is received and processed by the human cochlea. Conversion was done using `pycochleagram` (Gonzalez, 2018). Representations used 24 filter bands to capture frequencies from 50 to 8000 Hz (roughly one band per 1/3 octave), where frequencies captured by individual bands overlapped by 50%, along with a low-pass and high-pass filter. Conversion resulted in representations of shape 480 time steps x 26 filter bands. The time dimension was then removed by averaging over said dimension, resulting in 1x26 vectors.

4.3 Model Architecture and Training

While Keidel (2007) implemented a recurrent neural network, the current work implements feedforward neural networks. This change was made for two reasons: a time dimension is not necessary to represent monophthongal vowels, rendering recurrence unnecessary, and feedforward networks are simpler than recurrent networks, allowing the performance of models simpler than Keidel (2007)'s to be examined. Additionally the present models differ in training objective from Keidel (2007)'s model; Keidel (2007)'s model learned to produce acoustic feature vectors, while the present models learn to produce a single vowel label. This change was made so as to not restrict models' learned representations to a specific, provided feature set. Two types of models were trained: a Catalan baseline model, intended to simulate Catalan-dominant listeners (which produces the labels /i ɛ e a o ɔ u/) and Spanish models, intended to simulate Spanish-dominant listeners (which produce the labels /i e a o u/).

Models are four layer feedforward neural networks with two hidden layers[9]. ReLU non-linear activation is applied after both hidden layers. The input layer has 26 units (one unit for each filter band) and the output layer has 5 or 7 units (one

[7] A small portion of both corpora were unable to be aligned. Spanish: 2,194 files, Catalan: 102 files.

[8] Before alignment, probabilistic information was removed from Vargo (2025)'s dictionary as it contained both probabilistic and non-probabilistic entries.

[9] The performance of five layer networks with three hidden layers was also explored. The classification accuracy of five layer models on their L1 (Catalan for the baseline, Spanish for Spanish models), either did not improve or degraded relative to the accuracies of four layer models. As such, this work focuses on four layer models.

unit for each Spanish or Catalan vowel). This general architecture can be seen in Figure 1. Models were trained using cross entropy loss and the Adam optimizer (Kingma and Ba, 2015).

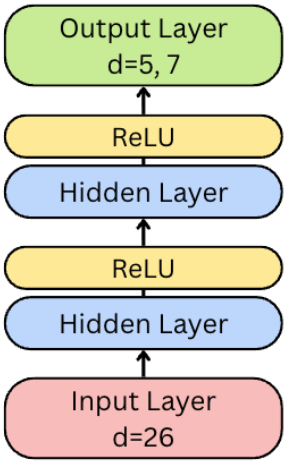


Figure 1: Model architecture.

Catalan Baseline. To provide a comparison for Spanish models, a Catalan baseline model was implemented. Given the small size of the Catalan training set, the model was implemented with a combination of grid search and 5-fold cross-validation. Grid search was conducted targeting the learning rate (0.01, 0.001, 0.0001), hidden layer dimension (7, 14, 26, 52)[10], number of training epochs (10, 20, 30, 40, 50, 60, 70), and batch size (4, 16, 32). 5-fold cross-validation was used to find optimal model parameters, which were then used to train a fresh initialization. Optimal parameters consisted of a 0.001 learning rate, 70 training epochs, a hidden dimension of 52, and a batch size of 4. These parameters achieved an average 66% accuracy during cross-validation.

Spanish Models. To find optimal parameters, grid search was conducted targeting the learning rate (0.01, 0.001, 0.0001), hidden layer dimension (5, 10, 26, 52), number of training epochs (10, 20, 30, 40, 50, 60, 70), and batch size (4, 16, 32). Models were trained on the Spanish training set, and their classification accuracy was evaluated on the Spanish development set.

Two models were chosen for evaluation. The first was the best model found during grid search. This model had parameters of 0.001 learning rate, 50 training epochs, a hidden dimension of 52, and a batch size of 4, and achieved 76.86% on the Spanish development set (henceforth Spanish-52). The second model was the highest performing model with a hidden dimension of 5 units (henceforth Spanish-5). This model had additional parameters of 0.01 learning rate, 30 training epochs, and a batch size of 16, and achieved 73.22% accuracy on the Spanish development set. An overview of evaluated models can be seen in Table 2.

	Spanish-5	Spanish-52	Baseline
Learning Rate	0.01	0.001	0.001
Training Epochs	30	50	70
Hidden Dimension	5	52	52
Batch Size	16	4	4
Accuracy	73.22%	76.86%	66%

Table 2: Overview of evaluated models.

4.4 Evaluation Metrics

Models were evaluated in two ways. First, model categorization of vowels specific to Catalan, namely /ɛ/ and /ɔ/, was measured. Given empirical results, Spanish models were expected to classify /ɛ/ as /e/ or /a/, and /ɔ/ as /o/, more frequently than the baseline (Pallier et al., 1997; Sebastián-Gallés and Soto-Faraco, 1999; Amengual, 2016).

Second, model discrimination was measured for three test contrasts, /e/~/ɛ/, /o/~/ɔ/, /ɛ/~/a/, along with three control contrasts /i/~/e/, /u/~/o/, /ɔ/~/a/. Discrimination was approximated by comparing a model's label distributions for two vowels within a contrast. For example, consider the /e/~/ɛ/ contrast. A Catalan model is expected to classify /ɛ/ and /e/ as two separate vowels, and as such, the labels assigned to /ɛ/ and /e/ should not have similar distributions. A Spanish model, however, is expected to classify /ɛ/ and /e/ as the same vowel, and as such, the labels assigned to /ɛ/ and /e/ should have similar distributions. Total variation distance (1) was used for this purpose, where P and Q represent probability distributions over a model's label set[11].

[10] 7 units corresponds to one unit for each vowel in the Catalan inventory; 26 units corresponds to one unit for each filter band in cochleagram representations.

[11] The baseline produces distributions over seven categories while Spanish models produce distributions over only five categories. This introduces a potential confound when comparing

$$\mathrm{TV}(\mathrm{P}, \mathrm{Q}) = \frac{1}{2} \sum_{x} |\mathrm{P}(x) - \mathrm{Q}(x)| \qquad (1)$$

Differences between model total variation distances were compared to Amengual (2016)'s findings. Catalan and Spanish models were expected to exhibit similar total variation distances on control contrasts, while Spanish models were expected to show lower total variation distances than the Catalan baseline on test contrasts.

5 Results

To determine whether models learned their L1, the baseline was evaluated on the Catalan testing set, and Spanish models were evaluated on the Spanish testing set. Heatmaps for the baseline, Spanish-5, and Spanish-52 appear in Figures 2, 3, and 4, respectively.

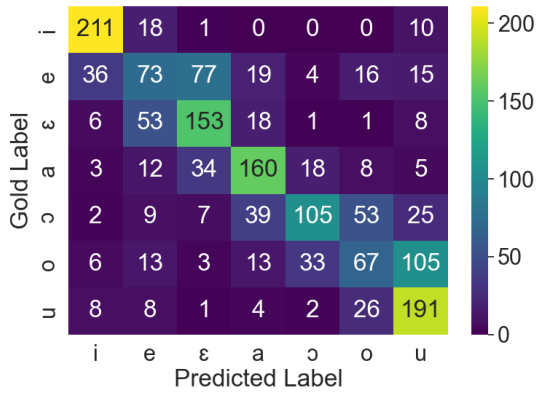


Figure 2: Baseline performance on Catalan testing set. Accuracy: 57.14%

Given the Catalan training set's small size, it is unsurprising that the baseline did not completely learn the Catalan inventory, that is, that a perfect diagonal does not appear in Figure 2. Despite this, the model still provides a comparison point for Spanish models; the baseline has knowledge of the two vowels specific to the Catalan inventory, /ɛ/ and /ɔ/, and the distribution of the baseline's predictions can be compared to those of the Spanish models.

Both Spanish models successfully learned the Spanish vowel inventory, as evidenced by the bright diagonals in Figures 3 and 4. These models can then be used to approximate Spanish-dominant listeners' perception of Catalan vowels.

Recall that the three Catalan contrasts of interest are /e/~/ɛ/, /o/~/ɔ/, and /ɛ/~/a/. Spanish models are expected to classify /ɛ/ as /e/ or /a/, and /ɔ/ as /o/. To evaluate Spanish models' classification of Catalan vowels, models were evaluated on the Catalan testing set. Spanish-5's classification heatmap can be seen in Figure 5 and Spanish-52's classification heatmap in Figure 6.

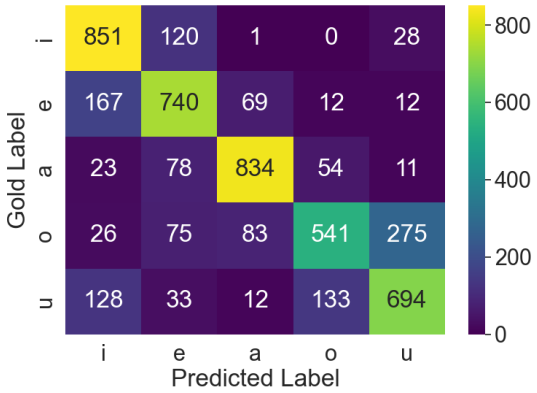


Figure 3: Performance of Spanish-5 on Spanish testing set. Accuracy: 73.2%

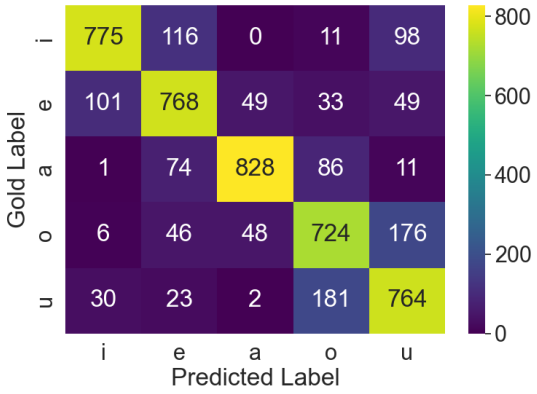


Figure 4: Performance of Spanish-52 on Spanish testing set. Accuracy: 77.18%

Figures 5 and 6 indicate that Spanish models largely maintain knowledge of vowels shared between the Spanish and Catalan inventories. Additionally, in line with predictions, /ɛ/ is most frequently classified as /e/, followed by /a/, and /ɔ/ is most frequently classified as /o/.

A more concrete evaluation of model classification can be achieved by investigating model classification percentages. Percentages within Table 3

between models, as total variation distances for the Catalan baseline may be artificially larger than Spanish models' differences. In practice, however, Catalan and Spanish models have similar total variation distance on control contrasts when hidden dimension is held constant.

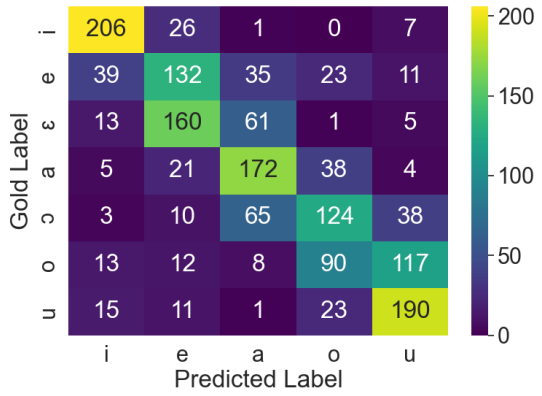


Figure 5: Performance of Spanish-5 on Catalan testing set. Accuracy: 47.02%

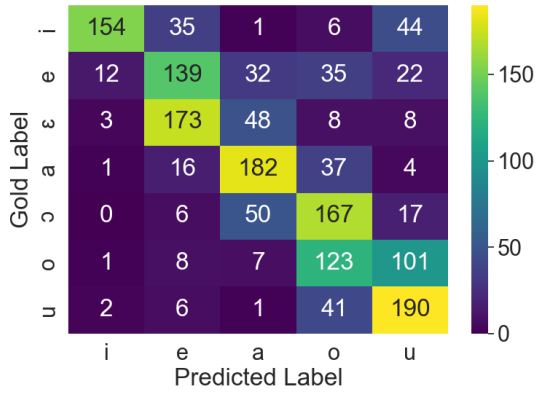


Figure 6: Performance of Spanish-52 on Catalan testing set. Accuracy: 46.9%

indicate that Spanish models did indeed classify /ɛ/ as /e/ or /a/, and /ɔ/ as /o/, more frequently than the baseline, with the difference in classification percentage being most pronounced for /ɛ/ as /e/, followed by /ɔ/ as /o/, and finally /ɛ/ as /a/. This demonstrates that Spanish models classified /ɛ/ and /ɔ/ differently than the Catalan baseline, and in ways that align with empirical results.

	/ɛ/ as /e/	/ɛ/ as /a/	/ɔ/ as /o/
Spanish-5	66.67%	25.42%	51.67%
Spanish-52	72.08%	20.0%	69.58%
Baseline	22.08%	7.5%	22.08%

Table 3: Frequency with which models classify /ɛ/ as /e/, /ɛ/ as /a/, and /ɔ/ as /o/.

Direct comparison to experimental results is available for one contrast: /e/~/ɛ/. Pallier et al. (1997) assessed Catalan-dominant and Spanish-dominant bilinguals' identification of a synthetic

	[e] as /ɛ/	[ɛ] as /ɛ/
Catalan-dominant	~15%	~82%
Spanish-dominant	~55%	~35%

Table 4: Pallier et al. (1997)'s classification percentages for stimulus 1 - [e], and stimulus 7 - [ɛ]. Percentages are estimated from Pallier et al. (1997) Figure 1.

vowel continuum spanning from [e] to [ɛ], and found that Catalan-dominant identifications formed a clear s-shaped curve, while Spanish-dominant identifications remained largely flat across the vowel continuum. Participant identifications of the first and last stimuli (prototypical [e] and [ɛ], respectively) as /ɛ/ can be seen in Table 4.

Using the percentages in Table 4, other identifications can be inferred; Catalan-dominant participants identified ~18% of [ɛ]s as /e/; Spanish-dominant participants identified ~65% of [ɛ]s as /e/. The current models mirror this pattern, with the Catalan baseline classifying 22.08% of [ɛ]s as /e/, and the Spanish models classifying upwards of 66.67% of [ɛ]s as /e/. This demonstrates that models' identification of Catalan /ɛ/ aligns with that of the population they intend to capture.

Models were also evaluated on their discrimination, which was approximated using total variation distance, and compared to empirical results. In an AXB task, Amengual (2016) found that Spanish-dominant listeners show comparable discrimination accuracy to Catalan-dominant listeners on Catalan's /i/~/e/, /u/~/o/, /ɔ/~/a/ contrasts, but significantly higher discrimination error than Catalan-dominant listeners on Catalan's /e/~/ɛ/ (Catalan-dominant 5%, Spanish-dominant 11%), /o/~/ɔ/ (Catalan-dominant 4%, Spanish-dominant 12.3%), /ɛ/~/a/ (Catalan-dominant 6.2%, Spanish-dominant 11.2%) contrasts. As such, Spanish models are expected to exhibit similar total variation distances to the baseline on /i/~/e/, /u/~/o/, /ɔ/~/a/, but lower total variation distances than the baseline on /e/~/ɛ/, /o/~/ɔ/, /ɛ/~/a/. Model total variation distances can be seen in Table 5.

Comparing the Catalan baseline and Spanish-52 partially yields the same results found by Amengual (2016); the models show similar total variation distances on the control contrasts, and Spanish-52 shows lower total variation distance than the baseline on one test contrast, namely /e/~/ɛ/.

Comparing the Catalan baseline and Spanish-5 does not yield the same results found by Amengual (2016); Spanish-5 shows lower total variation

	Test Contrasts			Control Contrasts		
	/e/~/ɛ/	/ɛ/~/a/	/o/~/ɔ/	/i/~/e/	/u/~/o/	/ɔ/~/a/
Spanish-5	0.22	0.62	0.38	0.7	0.31	0.5
Spanish-52	0.21	0.68	0.36	0.68	0.37	0.6
Baseline	0.32	0.69	0.42	0.73	0.37	0.63

Table 5: Total variation distance between model classification distributions.

distances than the baseline on test conditions as well as control conditions. So, while Spanish-5's classification of Catalan vowels is consistent with Spanish-dominant listeners' perception, this is not reflected in the model's discrimination.

In sum, Spanish models' classification differed from that of the Catalan baseline; Spanish-5 and Spanish-52 classified /ɛ/ as /e/ or /a/, and /ɔ/ as /o/ more frequently than the baseline. Additionally, while Spanish models largely did not exhibit expected discrimination results, Spanish-52 did show lower total variation distance on Catalan's /e/~/ɛ/ contrast than the baseline. Ultimately, these perceptual patterns exhibited by Spanish models are consistent with those demonstrated by adult Spanish-dominant listeners.

6 Discussion

In an effort to model Spanish-dominant listeners' cross-linguistic perception of Catalan vowels, this work trained small supervised feedforward neural networks to perform Spanish vowel classification and evaluated model classification of Catalan vowels. Vowels were extracted from Spanish and Catalan audio corpora, and processed to represent human auditory processing of vowels. While Spanish models largely did not exhibit expected discrimination patterns, Spanish models classified Catalan /ɛ/ as /e/ or /a/, and Catalan /ɔ/ as /o/, closely mimicking perceptual patterns exhibited by adult Spanish-dominant listeners.

The present work builds on Keidel (2007)'s recurrent neural model of cross-linguistic speech perception by mimicking L2 perceptual patterns using simpler networks trained and evaluated on larger amounts of data. Keidel (2007)'s model was a five layer recurrent neural network with hidden layers as large as 50 units, while the largest Spanish model had a comparable hidden dimension but a simpler, four layer feedforward architecture. Moreover, even a network with 5-unit hidden layers was able to mimic Spanish-dominant listeners' perception. Additionally, Keidel (2007)'s model was trained and evaluated on a limited amount of laboratory speech, while present models were trained and evaluated on vowels extracted from audio corpora. In short, this work demonstrates that models simpler than Keidel (2007)'s can predict L2 perceptual patterns, even given more complicated data.

This work also builds upon Matusevych et al. (2023) by showing that supervised models are capable of capturing Spanish-dominant listeners' perception of Catalan's /e/~/ɛ/ contrast. This suggests that learning style, instead of model size, is more important when modeling cross-linguistic speech perception, as Matusevych et al. (2023)'s large unsupervised models failed to predict Spanish-dominant listeners' perception of this contrast.

Additionally, this work demonstrates that computational models of cross-linguistic speech perception need not necessarily borrow from speech technologies. Recent modeling work has focused on using real speech, namely audio corpora, to train computational models of cross-linguistic speech perception. As such, model size has increased, with large speech models often being employed (Schatz and Feldman, 2018; Matusevych et al., 2023). The present work, however, illustrates that realistic input does not necessarily require large, specialized models; small feedforward networks are able to learn from audio corpora, and make predictions of cross-linguistic speech perception. This is advantageous, as small models are generally lighter to implement and easier to interpret than larger models. Small models also often make use of simpler mechanisms, which may be desirable from a cognitive perspective.

This work is, of course, not without limitations. One straightforward limitation lies in the baseline, specifically in the small amount of data used to train it. The baseline was likely under-trained, making it an imperfect point of comparison.

This work also implemented only a single model architecture, the feedforward neural network. Given this work's interest in small, simple models, the performance of approaches such as k-nearest

neighbors, k-means clustering, or logistic regression, should also be investigated.

A question raised by this work concerns model perception of /u/~/o/. All models classified /o/ as /u/ with some frequency[12]. This is interesting, given that balanced Spanish-Catalan bilingual infants have been shown to briefly lose the ability to discriminate the /u/~/o/ contrast (Bosch and Sebastián-Gallés, 2003). While the present work did not focus on this contrast, future work should more carefully consider model perception of /u/~/o/.

Finally, to explore the generalizability of this approach, the present work should be replicated on different segmental contrasts or language pairs. For example, an extension within the Spanish-Catalan pairing is the /ʃ/~/ʒ/ contrast, which is present in Catalan, but not in Spanish (Sebastián-Gallés and Soto-Faraco, 1999).

7 Acknowledgments

I am grateful to Bill Idsardi and Naomi Feldman for their valuable feedback during the course of this project. I also thank the anonymous reviewers for their helpful comments.

References

Frans Adriaans. 2024. Computational approaches to bilingual phonetics and phonology. In Mark Amengual, editor, *Bilingual Phonetics and Phonology*, pages 126–144. Cambridge University Press.

Mark Amengual. 2016. The perception and production of language-specific mid-vowel contrasts: Shifting the focus to the bilingual individual in early language input conditions. *International Journal of Bilingualism*, 20(2):133–152.

Catherine Best. 1995. A direct realist view of cross-language speech perception. In Winifred Strange, editor, *Speech Perception and Linguistic Experience: Issues in Cross-Language Research*, pages 171–204. Timonium, MD: York Press.

Catherine Best, Gerald McRoberts, and Elizabeth Goodell. 2001. Discrimination of non-native consonant contrasts varying in perceptual assimilation to the listener's native phonological system. *The Journal of the Acoustical Society of America*, 109(2):775–794.

Catherine Best and Michael Tyler. 2007. Nonnative and second-language speech perception: Commonalities and complementarities. In Ocke-Schwen Bohn and Murray Munro, editors, *Language experience in second language speech learning: In honor of James Emil Flege*, pages 13–34. John Benjamins.

Laura Bosch and Núria Sebastián-Gallés. 2003. Simultaneous bilingualism and the perception of a language-specific vowel contrast in the first year of life. *Language and Speech*, 46:217–243.

Alfonso Caramazza, Grace Yeni-Komshian, E Zurif, and E Carbone. 1973. The acquisition of a new phonological contrast: the case of stop consonants in french english bilinguals. *The Journal of the Acoustical Society of America*, 54(2):421–428.

Joan Carbonell and Joaquim Llisterri. 1999. Catalan. In *Handbook of the International Phonetic Association*, pages 61–65. Cambridge University Press.

James Flege. 1995. Second language speech learning: Theory, findings and problems. In Winifred Strange, editor, *Speech Perception and Linguistic Experience: Issues in Cross-Language Research*, pages 233–277. Timonium, MD: York Press.

Ray Gonzalez. 2018. pycochleagram.

James Keidel. 2007. *Behavioral, Modeling and Neuroimaging Studies of Auditory Expertise*. Ph.D. thesis, University of Wisconsin-Madison.

James Keidel, Jason Zevin, Keith Kluender, and Mark Seidenberg. 2003. Modeling the role of native language knowledge in perceiving nonnative speech contrasts. In *15th International Congress of Phonetic Sciences*, pages 2221–2224, Barcelona, Spain.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, San Diego, CA.

Oddur Kjartansson, Alexander Gutkin, Alena Butryna, Isin Demirsahin, and Clara Rivera. 2020. Open-source high quality speech datasets for Basque, Catalan and Galician. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages and Collaboration and Computing for Under-Resourced Languages*, pages 21–27, Marseille, France. European Language Resources association.

Tim Mahrt. 2016. PraatIO.

Yevgen Matusevych, Thomas Schatz, Herman Kamper, Naomi Feldman, and Sharon Goldwater. 2023. Infant phonetic learning as perceptual space learning: A crosslinguistic evaluation of computational models. *Cognitive Science*, 47(4):e13314.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal forced aligner: Trainable text-speech alignment using Kaldi. In *Interspeech 2017*, pages 498–502, Stockholm, Sweden.

[12]Baseline: 43.75%, Spanish-5 evaluated on Spanish: 27.5%, Spanish-52 evaluated on Spanish: 17.6%, Spanish-5 evaluated on Catalan: 48.75%, Spanish-52 evaluated on Catalan: 42.08%.

Michael McAuliffe and Morgan Sonderegger. 2022a. Spanish MFA acoustic model v2.0.0. Technical report.

Michael McAuliffe and Morgan Sonderegger. 2022b. Spanish (Spain) MFA G2P model v2.0.0. Technical report.

Christophe Pallier, Laura Bosch, and Núria Sebastián-Gallés. 1997. A limit on behavioral plasticity in speech perception. *Cognition*, 64(3):B9–B17.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. 2020. MLS: A large-scale multilingual dataset for speech research. In *Interspeech 2020*, pages 2757–2761. ISCA.

Joselyn Rodriguez, Kamala Sreepada, Ruolan Leslie Famularo, Sharon Goldwater, and Naomi Feldman. 2024. Self-supervised speech representations display some human-like cross-linguistic perceptual abilities. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, pages 458–463, Miami, FL, USA. Association for Computational Linguistics.

Rebecca Ronquest. 2018. Vowels. In Kimberly Geeslin, editor, *The Cambridge Handbook of Spanish Linguistics*, pages 145–164. Cambridge University Press.

Thomas Schatz and Naomi Feldman. 2018. Neural network vs. HMM speech recognition systems as models of human cross-linguistic phonetic perception. In *Conference on Cognitive Computational Neuroscience*, Philadelphia, Pennsylvania.

Thomas Schatz, Naomi H. Feldman, Sharon Goldwater, Xuan-Nga Cao, and Emmanuel Dupoux. 2021. Early phonetic learning without phonetic categories: Insights from large-scale simulations on realistic input. *Proceedings of the National Academy of Sciences*, 118(7):e2001844118.

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

Núria Sebastián-Gallés and Salvador Soto-Faraco. 1999. Online processing of native and non-native phonemic contrasts in early bilinguals. *Cognition*, 72(2):111–123.

Christian J. Steinmetz and Joshua D. Reiss. 2021. pyloudnorm: A simple yet flexible loudness meter in python. In *150th Audio Engineering Society Convention*.

Julian Vargo. 2025. Catalan montreal forced alignment bundle.

A Family of Effective Methods for Decompiling Canonical Acceptors, Instantiated for Languages of Dot-Depth One and Tier-Based Extensions

Dakotah Lambert
Department of Mathematics and Computer Science
Lake Forest College
dakotahlambert@acm.org

Abstract

Many kinds of logical systems have been employed in constructing formal languages to model phonological phenomena. A common theme among them is that the systems compile into finite automata. Two questions naturally arise. Can a given phenomenon be described with another logical system? And, if so, what is that description?

To the first question, algebraic techniques are well established through deep connections with logic and automata. To the second, the situation is less clear. Translations from automata are established for first-order and monadic second-order logics under precedence, but these may not translate easily to the simpler systems we often use. Translations for simple cases of restricted propositional logic (strictly local or strictly piecewise languages) are established, but insufficient to describe attested phenomena.

The present work establishes a general way to handle many systems in between. Specifically, we show how to translate between certain kinds of algebraic varieties $\mathbf{V}$ (systems defined by universally satisfied identities) and associated logical systems, then use decomposition to handle classes of the form $\mathbf{V} * \mathbf{D}$, where the notion of "symbol" is replaced by "k-block". With this, we handle several (unrestricted) propositional logics, facilitating logical description of natural language.

1 Introduction

Logical formalisms and their correspondence to finite automata provide a rich and well-grounded base for descriptions of natural language and linguistic theories (see Rogers, 1998; Hulden, 2009; Heinz, 2018, 2025). There are many tools that convert logical descriptions into finite automata, including MONA by Henriksen et al. (1995) focused on monadic second-order logic, Foma by Hulden (2009) focused on first-order logic, or the Language Toolkit by Lambert (2024) focused on quantifier-free systems of propositional logic. Compiling logical expressions to automata is solved. The present work addresses the inverse: decompiling (or "factoring") automata into human-readable logical expressions.

Algebraic techniques provide a universal approach to deciding whether a given automaton belongs to a given class, whether it can be written in a particular logical form (Lambert, 2026). The Language Toolkit provides these tools, but, in general, these techniques do not provide the logical description. The Büchi–Elgot–Trakhtenbrot theorem (Büchi, 1960; Elgot, 1961; Трахтенброт, 1961) provides a means to convert from automata back into monadic second-order logic. McNaughton and Papert (1971) show how to convert appropriate automata back into first-order logic with precedence (less-than). Below, for all but the weakest logics, less is available.

A language is locally testable in the strict sense (strictly local) if it is defined by a collection of forbidden substrings. A language is piecewise testable in the strict sense (strictly piecewise) if it is defined by a collection of forbidden subsequences. In either case, any presence of a forbidden factor immediately rules out the word. Rogers and Lambert (2019) provide mechanisms to factor automata into conjunctive systems of constraints forbidding substrings and/or subsequences, thereby handling these restricted propositional systems. Janoušek and Plachý (2024) provide an alternative approach for factoring strictly local systems. The vast space between here and first-order seems largely unexplored.

The present work addresses this gap. We provide a general approach, grounded in algebraic techniques, to convert from automata to a wide range of propositional logics. A Python implementation is provided as an appendix. Any algebraic class with a (parameterized) finite free object (see §2.3)

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 25–36
July 3-4, 2026 ©2026 Association for Computational Linguistics

and interpretable elements can be handled. By decomposing systems of the form $\mathbf{V} * \mathbf{D}$ (see §4), we generalize those results to systems wherein the notion of "symbol" is replaced by "k-block". A k-block is a contiguous substring of length k, the same kind of structure considered in an n-gram model.

This paper presents these methods and illustrates them with respect to the analysis of phonological patterns in Navajo, Karanga Shona, Tsuut'ina, Luganda, and stress patterns in the StressTyp2 database of Goedemans et al. (2015). These examples are chosen because they illustrate increasingly complex classes between strictly local or strictly piecewise and first-order logic. In particular, after establishing the basic method for content languages (Navajo, §3), this method is extended to the locally testable class (Karanga Shona, §4), the tier-based locally testable class (Tsuut'ina, §6), the locally threshold testable class (Luganda, §7), and the piecewise testable and dot-depth one classes (stress patterns, §8). Importantly, the core structure of the algorithm is identical throughout.

2 Background

2.1 Machines and Structures

Throughout this work, determinism will be assumed, so the word "deterministic" will be omitted outside of definitions. A **deterministic finite semiautomaton** is defined by a finite alphabet (Σ), a finite set (Q) of states, and a transition function that maps a letter–state pair to a state. Let $\delta_x(q)$ denote the application of this transition function to the input consisting of the letter $x \in \Sigma$ at the state $q \in Q$. This transition function is extended to words by $\delta^*_\lambda(q) = q$ and $\delta^*_{uv} = \delta^*_v \circ \delta^*_u$, where λ represents the unique word of length zero. A **deterministic finite-state acceptor** (DFA) is a deterministic finite semiautomaton augmented with an initial state ($q_0 \in Q$) and a set ($Q_f \subseteq Q$) of accepting states. We write these components as a five-tuple $A = \langle \Sigma, Q, \delta, q_0, Q_f \rangle$. A DFA defines a language L such that a word w is in L if, and only if, $\delta^*_w(q_0) \in Q_f$.

A state $q \in Q$ is **reachable** if there exists some word $w \in \Sigma^*$ such that $\delta^*_w(q_0) = q$). Two states q and q' in Q are **distinguishable** if there exists some word $w \in \Sigma^*$ such that $\delta^*_w(q) \in Q_f$ but $\delta^*_w(q') \notin Q_f$ or vice versa. An automaton is said to be in canonical form, or simply **canonical**, if all states are reachable and pairwise distinguishable.

A **deterministic finite-state transducer** (DFT) is a deterministic finite-state acceptor that associates computation with output. There is an output alphabet (Γ), and Γ^* is the set of words composed of letters in Γ. A DFT has a universal prefix ($\pi \in \Gamma^*$), a transition-output function (γ) where $\gamma_x(q) \in \Gamma^*$ is the output emitted upon encountering the letter $x \in \Sigma$ in state $q \in Q$, and $\sigma(q) \in \Gamma^*$ is the output emitted upon stopping in the state $q \in Q_f$. The γ function extends to words much like δ: $\gamma^*_\lambda(q) = \lambda$ and $\gamma^*_{xw}(q) = \gamma_x(q)\gamma^*_w(\delta_x(q))$. If w is a word such that $\delta^*_w(q_0) \in Q_f$, then the output associated with w is $\pi\gamma^*_w(q_0)\sigma(\delta^*_w(q_0))$, otherwise there is no such output.

A **semigroup** is a set alongside an associative operation. Function composition is associative; the set of δ^*_w for $w \in \Sigma^+$ is therefore a semigroup, where Σ^+ is the set of nonempty words composed of characters in Σ. This is called the **transition semigroup** of the semiautomaton. As these are functions from a finite domain to a finite codomain, there are only finitely many such functions, so every semiautomaton is associated with a finite semigroup. The transition semigroup of a DFA in some sense describes the structure of its associated language: if $\delta^*_u = \delta^*_v$, this means that there is a sense in which u and v have equivalent behavior. Namely, for any context $a__b$, it holds that $\delta^*_{aub} = \delta^*_b \circ \delta^*_u \circ \delta^*_a = \delta^*_b \circ \delta^*_v \circ \delta^*_a = \delta^*_{avb}$, so aub is accepted if, and only if, avb is accepted. Let $u \sim v$ denote this equivalence. Equivalence is maintained by concatenation; it is easy to verify that if $u \sim v$ and $x \sim y$, then $ux \sim uy \sim vy \sim vx$. The **syntactic semigroup** of a language L is the transition semigroup of a canonical automaton that represents L.

Let suff_k be the function that retrieves the k-suffix of a word; that is, $\mathrm{suff}_k(\lambda) = \mathrm{suff}_0(w) = \lambda$ for $w \in \Sigma^*$ and $\mathrm{suff}_k(ua) = \mathrm{suff}_{k-1}(u)a$ for $u \in \Sigma^*$ and $a \in \Sigma$. A DFT is **input strictly k-local** if $\delta^*_w(q_0) = \delta^*_{\mathrm{suff}_{k-1}(w)}(q_0)$ for all $w \in \Sigma^*$ (Chandlee et al., 2014). If $q \in Q$ is an arbitrary reachable state, then $q = \delta^*_v(q_0)$ for some $v \in \Sigma^*$. If the length of w is at least $k-1$, then $\mathrm{suff}_{k-1}(w) = \mathrm{suff}_{k-1}(vw)$ no matter what v is. From this we derive the following for any input strictly k-local transducer:

$$\begin{aligned}\delta^*_w(q) = \delta^*_{vw}(q_0) &= \delta^*_{\mathrm{suff}_{k-1}(vw)}(q_0) \\ &= \delta^*_{\mathrm{suff}_{k-1}(w)}(q_0) = \delta^*_w(q_0)\end{aligned}$$

That is, for sufficiently long strings, their corresponding transition functions are constant. This in

turn means that, so long as every state is reachable, we have not just that $\delta^*_w(q_0) = \delta^*_{\text{suff}_{k-1}(w)}(q_0)$, but in general $\delta^*_w = \delta^*_{\text{suff}_{k-1}(w)}$. In other words, for any string w in Σ^* it holds that $w \sim \text{suff}_{k-1}(w)$.

Lambert and Heinz (2023) point out that this corresponds to an algebraic property called "definiteness". A semigroup is **k-definite** if for any elements u and x_1, x_2, ..., x_k, it holds that $ux_1x_2 \dots x_k = x_1x_2 \dots x_k$. An input strictly k-local transducer has a transition semigroup that is $(k-1)$-definite. Let a and $b_1 \dots b_{k-1}$ be strings in Σ^+, and let $u = \delta^*_a$ and $x_i = \delta^*_{b_i}$ for $1 \leq i < k$. Because each b_i contributes at least one letter, we have the following.

$$\begin{aligned} ux_1 \dots x_{k-1} &= \delta^*_{ab_1 \dots b_{k-1}} \\ &= \delta^*_{\text{suff}_{k-1}(ab_1 \dots b_{k-1})} = \delta^*_{\text{suff}_{k-1}(b_1 \dots b_{k-1})} \\ &= \delta^*_{b_1 \dots b_{k-1}} = x_1 \dots x_{k-1} \end{aligned}$$

In terms of universal algebra, a **variety** is a collection of structures that satisfy a system of universally satisfied identities, where specific elements cannot be referenced, only variables. The k-definite semigroups satisfy $ux_1x_2 \dots x_k = x_1x_2 \dots x_k$ where the u and each x_i range over all elements; we write the corresponding variety as $\mathbf{D}_k = [\![ux_1x_2 \dots x_k = x_1x_2 \dots x_k]\!]$. This says that $\mathbf{D}_k$ is the set of semigroups that satisfy this identity. If the system were built from multiple identities, they would be written separated by semicolons.

2.2 Propositional Logics

We work with a classical truth-functional propositional logic, in which formulae are interpreted as being either true or false (contrasting with modal logics). Formulae are built from a collection of atoms and standard logical connectives of $\wedge$ (and), $\vee$ (or), and $\neg$ (not); quantifiers are not available.

Let Σ be a finite alphabet and $\mathcal{W}$ be (a model of) a word. We say $\mathcal{W} \vDash \phi$ (read "$\mathcal{W}$ satisfies ϕ") if the interpretation of the formula ϕ over $\mathcal{W}$ is true. We now define our atoms. Define $\text{content}(\mathcal{W})$ to be the set of symbols that occur in $\mathcal{W}$. Each symbol $x \in \Sigma$ is an atom, where $\mathcal{W} \vDash x$ means $x \in \text{content}(\mathcal{W})$. That is, $\mathcal{W} = uxv$ for some strings u and v. The atom $x_1x_2 \dots x_n$ (where each $x_i \in \Sigma$) represents occurrence as a **substring**: $\mathcal{W} \vDash x_1x_2 \dots x_n$ means $\mathcal{W} = ux_1x_2 \dots x_nv$ for some strings u and v. The atom $x_1 \,..\, x_2 \,..\, \cdots \,..\, x_n$ (where each $x_i \in \Sigma$) represents occurrence as a **subsequence**: $\mathcal{W} \vDash x_1 \,..\, x_2 \,..\, \cdots \,..\, x_n$ means that there exists some sequence $u_1, \dots, u_n$ of words of strictly increasing length such that $\mathcal{W} = u_ix_iv_i$ (for some v_i) for each x_i. This essentially means that the x_i occur in $\mathcal{W}$ in order, but not necessarily adjacently. This generalizes to subsequences of substrings: $\mathcal{W} \vDash x_{1.1} \dots x_{1.n} \,..\cdots..\, x_{m.1} \dots x_{m.n}$ means that the substrings $x_{i.1} \dots x_{i.n}$ occur in order, but not necessarily adjacently, with potential overlap. For example, we have that "ash" $\vDash$ as .. sh. Call this a **generalized subsequence**.

Let ϕ and ψ be logical formulae. The basic logical connectives are conjunction ($\phi \wedge \psi$, read "ϕ and ψ"), disjunction ($\phi \vee \psi$, read "ϕ or ψ"), and negation ($\neg\phi$, read "not ϕ"). We have that $\mathcal{W} \vDash \phi \wedge \psi$ if both $\mathcal{W} \vDash \phi$ and $\mathcal{W} \vDash \psi$, that $\mathcal{W} \vDash \phi \vee \psi$ if $\mathcal{W} \vDash \phi$ or $\mathcal{W} \vDash \psi$, or both, and that $\mathcal{W} \vDash \neg\phi$ if $\mathcal{W} \nvDash \phi$. A formula is in **disjunctive normal form** if it is a disjunction of one or more conjunctions of literals, where a literal is an atom or its negation. Our factorization systems provide formulae in disjunctive normal form.

2.3 Free Objects and Blockifiers

A **free** object in a class is the most generic form of a structure, where no equalities hold other than those implied by the definition of the class. For instance, Σ^+ is the free semigroup generated by Σ, as for any two sequences $u_1 \dots u_n$ and $v_1 \dots v_m$ of letters drawn from Σ, the resulting words in Σ^+ are equal if, and only if, $n = m$ and $u_i = v_i$ for $1 \leq i \leq n$. A free k-definite semiautomaton has a distinct state for each string of length up to k. The transition function is defined such that $\delta_a(w) = \text{suff}_k(wa)$.

We define a particular input $(k+1)$-strictly local transducer over this structure that we shall call a **$(k+1)$-blockifier**. The input alphabet shall be Σ, and the output alphabet shall consist of blocks of length $k+1$, potentially including boundaries:

$$\mathbb{B}_{k+1}(\Sigma) = \Sigma^{k+1} \cup (\rtimes \cdot \Sigma^k) \cup (\Sigma^k \cdot \ltimes) \cup (\rtimes \cdot \Sigma^{\leq k-1} \cdot \ltimes)$$

The states are those of the free k-definite semiautomaton: let w denote both a string and its corresponding state. Let the initial state q_0 be λ, indicating that no input has yet been seen, let the set of accepting states be $Q_f = Q$, and let the universal prefix π be λ. Consider a string u of length up to k and a letter $x \in \Sigma$. We have $\delta_x(u) = \text{suff}_k(ux)$, while the output functions depend on the length of

u, written $|u|$:

$$\gamma_x(u) = \begin{cases} \boxed{ux} & \text{if } |u| = k \\ \boxed{\rtimes ux} & \text{if } |u| = k-1 \\ \lambda & \text{otherwise} \end{cases}$$

$$\sigma(u) = \begin{cases} \boxed{u\ltimes} & \text{if } |u| = k \\ \boxed{\rtimes u\ltimes} & \text{otherwise} \end{cases}$$

This guarantees that every output symbol is either a fully bounded word whose length, including boundaries, is at most $k+1$, or it is an optionally bounded block of size exactly $k+1$.

The k-blockifier maps an input word w to the sequence of k-blocks in w in order from left to right, with overlap. For instance, the 3-blockifier applied to the word "abacaba" would emit $\boxed{\rtimes ab}\boxed{aba}\boxed{bac}\boxed{aca}\boxed{cab}\boxed{aba}\boxed{ba\ltimes}$. Throughout, let $B_k(w)$ be this function, mapping w through the k-blockifier to this output.

The blockifier is a transducer built with the structure of a free k-definite semigroup. Other free objects will be instrumental to our factorization methods.

3 Content Languages

Heinz (2010), citing Sapir and Hoijer (1967), analyzes a symmetric sibilant harmony system in Navajo, wherein the anteriority of sibilants is influenced by that of the rightmost sibilant. The surface constraint that arises from such a system is that a [+anterior] sibilant like 's' cannot occur in the same word as a [−anterior] sibilant like 'ʃ'. The acceptability of a word with respect to this constraint depends solely on its content. This means that, for this constraint, the order in which symbols occur is irrelevant ($uv \sim vu$ for all u and v), as is their quantity ($uu \sim u$ for all u). We call such a language a **content-language**. They can be described in a propositional logic whose only atoms are of the form $x \in \Sigma$. The variety $[\![uv = vu; uu = u]\!]$ is called $\mathbf{J}_1$ (see Almeida, 1995).

Eilenberg's Theorem (1976) associates with each variety $\mathbf{V}$ of semigroups a corresponding family $\mathcal{V}$ of languages: a language over Σ belongs to $\Sigma\mathcal{V}$ if, and only if, its syntactic semigroup belongs to $\mathbf{V}$. This family is thus also called a variety. For a language in $\Sigma\mathcal{J}_1$, by imposing an order on the symbols of the alphabet, words have a normal form reached by repeatedly using the $uv \sim vu$ rule to sort the symbols, then repeatedly using the $uu \sim u$ rule to deduplicate them. Equivalence is maintained throughout; we have $w \sim \text{content}(w)$. For a finite alphabet Σ there are only finitely many such forms. This fact indicates that the free object in $\Sigma\mathcal{J}_1$ is finite: there is one element per subset of Σ.

So a language L over Σ is in $\Sigma\mathcal{J}_1$ if, and only if, its syntactic semigroup is in $\mathbf{J}_1$. This does not, however, tell us what formula describes the language. But because the free object for $\Sigma\mathcal{J}_1$ is finite, we can extract this information as follows. Let $A = \langle \Sigma, Q, \delta, q_0, Q_F \rangle$ be the canonical acceptor for L. We maintain a stack of pairs of the form $\langle q, F \rangle$ where $q \in Q$ is a state in A and $F \subseteq \Sigma$ is an element of our free object. (This corresponds to a depth-first search. Replacing the stack with a queue performs a breadth-first search instead. Both are equally viable.) We also maintain a visited set of such pairs, a set of acceptable symbol sets and a set of rejected symbol sets. (In general, these are acceptable and rejected elements of the free object.) Let $R_\lambda(F) = F$ and $R_x(F) = F \cup \{x\}$, representing the transition function in a semiautomaton derived from our free object, and let $\alpha = \varnothing$, representing what would be the initial state in an acceptor so derived.

1. Begin with an empty visited set and a stack consisting of the single item $\langle q_0, \alpha \rangle$.
2. While the stack is nonempty:
   (a) Pop $\langle q, F \rangle$ from the stack and add it to the visited set.
   (b) For each alphabet symbol $x \in \Sigma$, push $\langle \delta_x(q), R_x(F) \rangle$ if it is unvisited.
   (c) If $q \in Q_f$, accept F.
   (d) If $q \notin Q_f$, reject F.

This process builds up the necessary portion of the free $\Sigma\mathcal{J}_1$ object, essentially unwrapping the canonical acceptor onto it. At the end of this process, the collection of acceptable symbol sets is interpretable as a formula in disjunctive normal form defining L.

The translation is as follows. Let $S = \{x_1, \ldots, x_m\}$ be an acceptable symbol set and let $\{y_1, \ldots, y_n\} = \Sigma - S$ be the symbols not in that set. Words that arrive at this element are those where each x_i appears and no y_i appears:

$$(x_1 \wedge \cdots \wedge x_m \wedge \neg y_1 \wedge \cdots \wedge \neg y_n)$$

Words are accepted if, and only if, they arrive at any such element, so the logical formula is the disjunction of the formulae derived from all acceptable symbol sets. In general, this approach produces

a logical formula for any variety with a finite free object whose elements are interpretable like this.

4 Blockifiers and Local Testability

A language is **locally k-testable** if the acceptability of a word w depends only on its set of substrings of length k, including word boundaries (McNaughton and Papert, 1971). Whether a shorter word of length less than $k-2$ is included is entirely arbitrary. This means that the acceptability of $w \in \Sigma^*$ is determined by $\mathrm{content}(B_k(w))$. A language is **locally testable** if, and only if, it is locally k-testable for some finite k.

So a language L is locally testable if, and only if, there is some language in $\mathbb{B}_k(\Sigma)\mathcal{J}_1$ that accepts $B_k(w)$ for all those w in L and rejects $B_k(w)$ for all those w not in L. That is, there is some DFA M whose transition semigroup is in $\mathbf{J}_1$ such that L is the language of $M \circ B_k$. In algebraic terms, the operator corresponding to this composition is the wreath product. The variety $\mathbf{V} * \mathbf{D}_{k-1}$ corresponds to the class of languages that can be defined by a composition of B_k into a DFA whose transition semigroup is in $\mathbf{V}$ (Straubing, 1985).

One example of a linguistically relevant locally testable pattern is the surface pattern of tone on non-assertive verb stems in Karanga Shona, a Bantu language described by Odden (1984) and analyzed by Jardine (2020) when defining the class of melody-local languages. Per Jardine (2020), there are two basic kinds of roots which pattern as follows, where superscript $+$ indicates one or more occurrences of the symbol it is attached to:

- H-toned: $\{\mathrm{H}, \mathrm{HL}, \mathrm{HLH}, \mathrm{HHLH}\} \cup \mathrm{HHHL}^+\mathrm{H}$
- L-toned: $\{\mathrm{L}, \mathrm{LH}, \mathrm{LHL}\} \cup \mathrm{LHHL}^+$

The thirteen-state canonical DFA recognizing the union of these two sets is shown in Figure 1, with the thirteenth, rejecting, sink state omitted. Each circle is a state, those with doubled outlines are accepting, and the transition function is indicated by overlaying the letter on an arrow from the source state to its destination. We assert that this is locally testable, and will verify this via factorization.

If we have an automaton $A = \langle \Sigma, Q, \delta, q_0, Q_f \rangle$ whose language we know to be locally testable, then we know that there must be some $k \geq 1$ for which this language is in $\mathbf{J}_1 * \mathbf{D}_{k-1}$. By fixing a correct value for k, we can iteratively build up the free object in $\mathbb{B}_k(\Sigma)\mathcal{J}_1$ much like we did in the previous

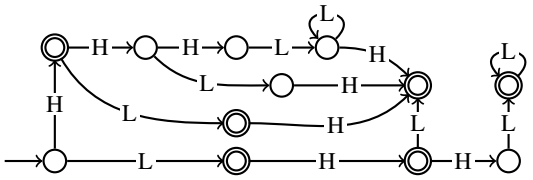


Figure 1: Karanga Shona non-assertive verb stems.

section, except that its input is not drawn from the acceptor A but from the output of the k-blockifier B_k as it is traced alongside A. We extend our stack to track not pairs but triples, consisting of an A-state, a B_k-state, and the corresponding set of output symbols (k-blocks). We still maintain a visited set, an acceptable set, and a rejected set. Still let $R_\lambda(F) = F$ and $R_x(F) = F \cup \{x\}$ otherwise, and $\alpha = \varnothing$ as we are working over the same free object structure.

1. Begin with an empty visited set and a stack consisting of the single item $\langle q_0, \lambda, \alpha \rangle$.
2. While the stack is nonempty:
   (a) Pop $\langle q, w, F \rangle$; mark it visited.
   (b) Push $\langle \delta_x(q), \mathrm{suff}_{k-1}(wx), R_{\gamma_x(w)}(F) \rangle$ for each symbol x, if not visited.
   (c) If $q \in Q_f$, accept $R_{\sigma(w)}(F)$.
   (d) If $q \notin Q_f$, reject $R_{\sigma(w)}(F)$.

This process builds up the necessary portion of the free $\mathbf{J}_1$ machine generated by k-blocks. Each acceptable set generated will have one prefix and one suffix (which may be merged as a fully anchored word) and zero or more factors without boundaries. The corresponding formula is the same disjunctive normal form as for content-languages, except that the atoms derived from symbols (which represent k-blocks) are interpreted as the substrings that they represent.

By this process, fixing $k = 4$, the following fifteen factor sets are emitted as valid for Karanga Shona non-assertive verb stems. Thirteen factor sets capture a single word apiece; these are $B_4(w)$ for w selected from H, L, HL, LH, HLH, LHL, HHLH, LHHL, LHHLL, HHHLH, HHHLLH, LHHLLL, or HHHLLLH. The remaining two have components that can be overlaid to produce arbitrarily long words, these are $\{\rtimes$LHH, LHHL, HHLL, HLLL, LLLL, LLL$\ltimes\}$, representing words of the form $\mathrm{LHHLLLL}^+$, and $\{\rtimes$HHH, HHHL, HHLL, HLLL, LLLL, LLLH, LLH$\ltimes\}$, representing words of the form $\mathrm{HHHLLLL}^+\mathrm{H}$.

5 Rejection or Approximation

If a language is not locally k-testable, then there must be two strings, u and v, which have the same set of k-blocks, but one is accepted while the other is rejected. This situation is detectable: if during the above process a factor set F is marked as acceptable when it has already been marked as rejected or vice versa, this indicates that the language is not locally k-testable and that some pair of wrongly indistinguishable u and v share factor set F. There are a few options at this point. On the one hand, one could choose to terminate the procedure and simply declare that the language is not locally k-testable and no factorization can exist. On the other hand, one could choose to continue as normal: by caring only about acceptable elements, we would say that every word w that shares a factor set with an acceptable word w' will be accepted, even if it should not be. Words whose factor set is never accepted will continue to be rejected. This is the smallest locally k-testable language that is a superset of the target language, a tight approximation. Flipping this, we can get the largest locally k-testable language that is a subset of the target language by rejecting all rejected elements. We can insist on correct factorization, or we can give a tight upper- or lower-bound.

This sensitivity allows factorization of arbitrary locally testable languages, even if we do not know the parameter k in advance. One way is to simply begin with the assumption that $k = 1$. If factorization fails because the language is not locally 1-testable, then try again with $k = 2$, and so on, increasing k by one each time. The result will be a factorization with the minimal possible value for k. This is the method that was used for Karanga Shona non-assertive verb stems.

Another approach is to use a theorem of Straubing (1985) to find a k that is guaranteed to work. Given two semigroup elements x and y, we say that $x \leq_{\mathcal{R}} y$ if, and only if, $x = y$ or there is some s such that $x = ys$. Similarly, we say that $x \leq_{\mathcal{L}} y$ if, and only if, $x = y$ or there is some s such that $x = sy$. The strict versions of these inequalities are defined such that $x <_{\mathcal{R}} y$ means $x \leq_{\mathcal{R}} y$ but $y \not\leq_{\mathcal{R}} x$ and similarly for $<_{\mathcal{L}}$. A chain under an inequality, $<$, is a sequence $x_1 < x_2 < \cdots < x_n$. Per Straubing (1985, Theorem 5.3), if a language is in $\mathbf{V} * \mathbf{D}_k$ for some (potentially unknown) k and c is the length of the longest chain under $<_{\mathcal{R}}$ or under $<_{\mathcal{L}}$, whichever is shorter, then that language is in $\mathbf{V} * \mathbf{D}_c$. As there are only finitely many elements in the syntactic semigroup of any regular language, this is effectively calculable. However, this does not necessarily yield the minimal value! Using this approach would have resulted in factoring with $k = 6$ instead of the minimal $k = 4$.

6 Nonsalient Symbols and Tier Analysis

Often in natural language patterns, we encounter dependencies that are nonlocal when considering a word as a whole, but become local when looking only at a particular class of symbols. The asymmetric sibilant harmony of Tsuut'ina reported by Cook (1978a,b) and analyzed by Heinz (2010) is one such case, where a [−anterior] sibilant may not follow a [+anterior] sibilant at any distance, but the reverse situation is allowed: /si-tʃiz-aʔ/ "my duck" surfaces as **ʃítʃídz**à, for example. Unlike the symmetric harmony of Navajo, this is not a content-language. Heinz (2010) analyzes this pattern in two different ways. A precedence based constraint could forbid the subsequence s…ʃ (but allow ʃ…s), Or a tier-based analysis could ignore symbols other than sibilants and forbid the substring sʃ on the result. The latter approach is expanded by Heinz et al. (2011) and explored by Lambert (2023).

We can effect such analyses by taking advantage of a key insight of Lambert (2023): nonsalient symbols (here, non-sibilants) are those that never change state. They are the symbols that label self-loops on every state. To analyze such a pattern, first remove the nonsalient symbols, then do the factorization on the projected result, and state that the resulting constraints apply on a tier. Here, if we assume a limited alphabet such as $\Sigma = \{\text{a}, \text{s}, \text{ʃ}\}$ we would use $k = 2$ and the tier $\text{T} = \{\text{s}, \text{ʃ}\}$. Running the analysis, the permissible factor sets on the tier T turn out to be $\{\rtimes\ltimes\}$, $\{\rtimes\text{s}, \text{s}\ltimes\}$, $\{\rtimes\text{ʃ}, \text{ʃ}\ltimes\}$, $\{\rtimes\text{ʃ}, \text{ʃs}, \text{s}\ltimes\}$, $\{\rtimes\text{ʃ}, \text{ʃʃ}, \text{ʃ}\ltimes\}$, $\{\rtimes\text{s}, \text{ss}, \text{s}\ltimes\}$, $\{\rtimes\text{ʃ}, \text{ʃʃ}, \text{ʃs}, \text{s}\ltimes\}$, $\{\rtimes\text{ʃ}, \text{ʃs}, \text{ss}, \text{s}\ltimes\}$, and $\{\rtimes\text{ʃ}, \text{ʃʃ}, \text{ʃs}, \text{ss}, \text{s}\ltimes\}$. In no case is sʃ permitted.

7 Multiplicity and Threshold Testability

The set of symbols in a word w is $\text{content}(w)$. Define $\text{content}_t(w)$ to be the *multiset* of symbols in w, counted up to some threshold t, beyond which further occurrences cannot be distinguished. If $uv\ vu$ and $uu^t\ u^t$, then we have $w \sim \text{content}_t(w)$: repeatedly apply the first to sort symbols, then repeatedly apply the second to deduplicate as much as possible. These identities define the variety $\mathbf{Acom}_t = [\![uv = vu; uu^t = u^t]\!]$. There is no finite free ob-

ject for $\mathbf{Acom}$ in general: such an object must be able to make all possible distinctions in terms of count but there are infinitely many natural numbers to distinguish. However, the family of parameterized classes satisfies $\mathbf{Acom}_1 \subseteq \mathbf{Acom}_2 \subseteq \cdots$. Extraction of $\mathbf{Acom}_t$ proceeds like that of $\mathbf{J}_1$ as seen in §3 Let $R_x(F)$ insert x into F, increasing its occurrence-count by one if it is not already t and let α be the empty multiset. If the parameter t is not known in advance, it can be derived from the procedure of §5: begin with $t = 1$, and if this fails, try again with increasingly large values for t. As each increase yields a superclass, if the system is $\mathbf{Acom}$ at all, this will eventually succeed.

The **locally threshold-t k-testable** languages are the generalization from symbols to k-blocks here, $\mathbf{Acom}_t * \mathbf{D}_{k-1}$, where the acceptability of a word is determined by its multiset of k-blocks, counted with multiplicity up to some threshold t, beyond which additional occurrences are not distinguished (see McNaughton and Papert, 1971 or Beauquier and Pin, 1989). A language is **locally threshold testable** if, and only if, it is locally threshold-t k-testable for some finite k and t. The $t = 1$ case is local testability; $k = 1$ is $\mathcal{Acom}$.

By Straubing's Theorem (1985), k is bounded above by the maximal length of a chain under $<_{\mathcal{R}}$ or $<_{\mathcal{L}}$, so we can use whichever is shorter. We can then either find t analytically, by building all products of k semigroup elements and for each such product p finding the smallest value t_p where $p^{t_p} = pp^{t_p}$, then selecting the largest of the resulting t_p. Or we can run the blockified factorization procedure of §4, beginning from $t = 1$ and increasing this value, as above, if inconsistencies arise.

One locally threshold testable pattern that arises in natural language is culminativity of stress, the constraint that at most one syllable with primary stress occurs (Hyman, 2014). This constraint belongs to many classes, in fact. For instance, it is tier-based locally testable (in the strict sense): if p represents primary stress and u represents lack of stress, then the 2-block pp is forbidden on the tier $\mathrm{T} = \{p\}$. It belongs to $\mathcal{Acom}$, as the multiset of symbols suffices to determine whether a word is acceptable, but Straubing's Theorem (1985) gives an upper bound of $k = 2$. Using $k = 2$ and $t = 1$, factorization fails; the pattern is not locally 2-testable. Using $k = 2$ and $t = 2$, the system returns the following collection of fourteen valid 2-multisets: $\{\rtimes\ltimes\}$, $\{\rtimes u, u\ltimes\}$, $\{\rtimes p, p\ltimes\}$, $\{\rtimes u, uu, u\ltimes\}$, $\{\rtimes u, up, p\ltimes\}$, $\{\rtimes p, pu, u\ltimes\}$, $\{\rtimes u, uu, uu, u\ltimes\}$, $\{\rtimes u, uu, up, p\ltimes\}$, $\{\rtimes u, up, pu, u\ltimes\}$, $\{\rtimes p, pu, uu, u\ltimes\}$, $\{\rtimes u, up, pu, uu, u\ltimes\}$, and $\{\rtimes u, uu, up, pu, uu, u\ltimes\}$. The block pp is never permissible, nor is any configuration in which pu or up appears twice. Finally, if a word starts with p it cannot contain up and if a word ends with p it cannot contain pu.

Another locally threshold testable constraint is unbounded high-tone plateauing, discussed by Jardine (2020) in relation to Luganda, in which tones may be low or high but surface forms only have a single span of high tones. That is, they may not contain a substring of the form $\mathrm{HL}^+\mathrm{H}$. Straubing's Theorem (1985) gives an upper bound of $k = 4$, but the system can be factored with $k = t = 2$. The resulting set of 34 valid 2-multisets serve to forbid three constructs. Words cannot start with H and contain LH. Words cannot end with H and contain HL. And words cannot contain HL twice.

8 Subsequences and Dot-Depth One

The piecewise testable languages were studied by Simon (1975). A language is **piecewise j-testable** if the acceptability of a word depends only on the set of j-subsequences in the word. A language is **piecewise testable** if it is piecewise j-testable for some finite j. The algebraic variety that corresponds to this class is typically known as $\mathbf{J}$. Like with $\mathbf{Acom}$, there is not a finite free object for $\mathbf{J}$, but $\mathbf{J}_1 \subsetneq \mathbf{J}_2 \subsetneq \mathbf{J}_3 \subsetneq \cdots$ and each $\mathbf{J}_j$ does have a finite free object. The free object for $\mathbf{J}_j$ has elements that correspond to possible sets of subsequences of length up to j, where if u is in the set and x is a subsequence of u, then x too is in the set. For such a system, the initial element is $\alpha = \{\lambda\}$ and $R_x(F) = F \cup \{\mathrm{suff}_j(sx) : s \in F\}$.

The dot-depth hierarchy was introduced by Cohen and Brzozowski (1971). A language has **dot-depth one** when the acceptability of a word depends on its set of j-subsequences of k-blocks, for some finite j and k. In other words, the class of languages of dot-depth one is $\mathbf{J} * \mathbf{D}$ (Knast, 1983; Straubing, 1985). The restriction to $j = 1$ yields the locally k-testable languages, and the restriction to $k = 1$ yields the piecewise j-testable languages.

Rogers and Lambert (2019) analyze and factor the patterns with associated automata in the StressTyp2 database of Goedemans et al. (2015). A result of this is that all but six of the properly subregular patterns factor into a combination of constraints that are locally testable in the strict sense, the com-

plement of such a constraint (which remains locally testable), or piecewise testable in the strict sense. The remaining six patterns have additional locally testable constraints that could not be extracted automatically but were instead supplied manually. However, this means that all of these properly subregular patterns are the combination of a locally testable component and a piecewise testable component; they have dot-depth one. The factorization methodology we present can fully factor all of these patterns, obviating the need for manual analysis even in these complex systems.

9 Conclusions

We presented a family of effective methods for extracting grammars of classes of the form $\mathbf{V} * \mathbf{D}$ from a canonical DFA. The block size k may be fixed or known in advance, or an overestimate can be derived by taking the length of the longest chain under $<_{\mathcal{R}}$ or $<_{\mathcal{L}}$, whichever is shorter. Alternatively, you can begin by assuming $k = 1$ and try again if necessary with increasing large k.

One selling point of the Language Toolkit of Lambert (2024) is that an analyst can describe a language in whatever terms are most convenient, then determine whether the language so-described is representable in various logical systems. We add to this now the capability to determine what that alternative representation actually is, at least for anything that is a content-language ($\mathbf{J}_1$), a content_t-language ($\mathbf{Acom}$), a piecewise j-testable language ($\mathbf{J}_j$), or the result of composing an input strictly k-local transducer into such a system, namely a locally k-testable language ($\mathbf{J}_1 * \mathbf{D}$), a locally threshold-$t$ k-testable language ($\mathbf{Acom} * \mathbf{D}$), or a language with dot-depth one ($\mathbf{J} * \mathbf{D}$), or the (single) tier-based extension of any of these.

In general, given a variety $\mathbf{V}$ with a finite free object, you need to define three things: you need a function R that maps an element of the free object and a symbol to another element of the free object, you need an initial element α, and you need a logical interpretation of the elements of the free object. From there, the procedure is as follows. Maintain a stack (or queue) of triples consisting of the current state in the DFA, the current state in a free input strictly k-local transducer, and the current element of the free object. Maintain also a set of visited triples, a set of acceptable elements of the free object, and a set of its rejected elements.

1. Begin with an empty visited set and a stack consisting of the single item $\langle q_0, \lambda, \alpha \rangle$.
2. While the stack is nonempty:
   (a) Pop $\langle q, w, F \rangle$; mark it visited.
   (b) Push $\langle \delta_x(q), \text{suff}_{k-1}(wx), R_{\gamma_x(w)}(F) \rangle$ for each symbol x, if not visited.
   (c) If $q \in Q_f$, accept $R_{\sigma(w)}(F)$.
   (d) If $q \notin Q_f$, reject $R_{\sigma(w)}(F)$.
3. Replace symbol k-blocks by their meaning.

If your desired variety $\mathbf{V}$ does not have a finite free object, but it can be parameterized into a family of subvarieties $\mathbf{V}_1 \subsetneq \mathbf{V}_2 \subsetneq \mathbf{V}_3 \subsetneq \cdots$, where each $\mathbf{V}_i$ does have such an object, then all is not lost. Like with the block size, it is possible to begin with the assumption that the parameter is 1, then try again with increasingly large values if that fails. If ever an element is marked as both acceptable and rejected, then either the language is not a member of $\mathbf{V} * \mathbf{D}$ or some of the parameters are too small.

For locally testable factorization, $\alpha = \varnothing$ and $R_z(F) = F \cup \{z\}$. For locally threshold-t testable systems, these parameters are the same, except that $\varnothing$ and F are not sets but t-multisets. For dot-depth one, $\alpha = \{\lambda\}$ and $R_z(F) = F \cup \{\text{suff}_j(sz) : s \in F\}$. In the end, the process will return valid sets of k-blocks, valid t-multisets of k-blocks, or valid sets of j-subsequences of k-blocks, respectively.

We demonstrated the utility of these techniques by factoring several phenomena from a range of phonological domains, including long-distance dependencies that arise in harmony, stress, and tone. We improved upon the results of Rogers and Lambert (2019), who factored the patterns in the StressTyp2 database but needed to supply additional constraints manually for six properly subregular patterns. Each of those six has dot-depth at most one, so the techniques demonstrated herein can be applied to avoid this burden. Similarly, we improved over the work of Janoušek and Plachý (2024), who presented a different method of factoring patterns that are locally testable in the strict sense. We maintain the ability to produce tight superset (or subset) approximations.

By stripping nonsalient symbols, performing the analysis, then restating the result in terms of a projective tier, this family of factorization methods extends trivially to the (single) tier-based extensions of factorable classes. Reasonable methods for handling multiple tiers remain an open problem.

Other future directions include searching for better parameter-finding methods and performing more analyses on more patterns in more languages.

References

Jorge Almeida. 1995. *Finite Semigroups and Universal Algebra*, volume 3 of *Series in Algebra*. World Scientific, Singapore.

Danièle Beauquier and Jean-Éric Pin. 1989. Factors of words. In Giorgio Ausiello, Mariangiola Dezani-Ciancaglini, and Simonetta Ronchi Della Rocca, editors, *Automata, Languages and Programming: 16th International Colloquium*, volume 372 of *Lecture Notes in Computer Science*, pages 63–79. Springer Berlin / Heidelberg.

Julius Richard Büchi. 1960. Weak second-order arithmetic and finite automata. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 6(1–6):66–92.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2014. Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics*, 2:491–503.

Rina S. Cohen and Janusz Antoni Brzozowski. 1971. Dot-depth of star-free events. *Journal of Computer and System Sciences*, 5(1):1–16.

Eung-Do Cook. 1978a. Palatalizations and related rules in Sarcee. In Eung-Do Cook and Jonathan Kaye, editors, *Linguistic Studies of Native Canada*, pages 19–35. University of British Columbia Press, Vancouver, Canada.

Eung-Do Cook. 1978b. The synchronic and diachronic status of Sarcee γ^y. *International Journal of American Linguistics*, 44(3):192–196.

Samuel Eilenberg. 1976. *Automata, Languages, and Machines*, volume B. Academic Press, New York, New York.

Calvin C. Elgot. 1961. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society*, 98(1):21–51.

R. W. N. Goedemans, Jeffrey Heinz, and Harry van der Hulst. 2015. StressTyp2.

Jeffrey Heinz. 2010. Learning long-distance phonotactics. *Linguistic Inquiry*, 41(4):623–661.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. In Larry Hyman and Frank Plank, editors, *Phonological Typology*, volume 23 of *Phonetics and Phonology*, chapter 5, pages 126–195. Mouton de Gruyter.

Jeffrey Heinz. 2025. The logical structure of phonological generalizations. *Phonological Studies*, 28:69–80.

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: Short Papers*, volume 2, pages 58–64, Portland, Oregon. Association for Computational Linguistics.

Jesper Gulmann Henriksen, Jakob Jensen, Michael Jørgensen, Nils Klarlund, Robert Paige, Theis Rauhe, and Anders Sandholm. 1995. MONA: Monadic second-order logic in practice. Technical Report RS-95-21, BRICS, Department of Computer Science, University of Aarhus, Aarhus, Denmark.

Mans Hulden. 2009. *Finite-State Machine Construction Methods and Algorithms for Phonology and Morphology*. Ph.D. thesis, The University of Arizona.

Larry M. Hyman. 2014. Do all languages have word accent? In Harry van der Hulst, editor, *Word stress: Theoretical and typological issues*, pages 56–82. Cambridge University Press.

Jan Janoušek and Štěpán Plachý. 2024. Shortest characteristic factors of a deterministic finite automaton and computing its positive position run by pattern set matching. In *SOFSEM 2024: Theory and Practice of Computer Science*, volume 14519 of *Lecture Notes in Computer Science*, pages 326–339. Springer.

Adam Jardine. 2020. Melody learning and long-distance phonotactics in tone. *Natural Language & Linguistic Theory*, 38:1145–1195.

Robert Knast. 1983. A semigroup characterization of dot-depth one languages. *RAIRO – Informatique théorique*, 17(4):321–330.

Dakotah Lambert. 2023. Relativized adjacency. *Journal of Logic, Language and Information*, 32(4):707–731.

Dakotah Lambert. 2024. System description: A theorem-prover for subregular systems: The Language Toolkit and its interpreter, plebby. In *Functional and Logic Programming: 17th Annual Symposium, FLOPS 2024*, volume 14659 of *Lecture Notes in Computer Science*, pages 311–328, Kumamoto, Japan. Springer, Singapore.

Dakotah Lambert. 2026. Multitier phonotactics with logic and algebra. *Phonology*, 43:e9 1–31.

Dakotah Lambert and Jeffrey Heinz. 2023. An algebraic characterization of total input strictly local functions. In *Proceedings of the Society for Computation in Linguistics*, volume 6, pages 25–34, Amherst, Massachusetts.

Robert McNaughton and Seymour Aubrey Papert. 1971. *Counter-Free Automata*. MIT Press.

David Odden. 1984. Stem tone assignment in Shona. In George N. Clements and John Goldsmith, editors, *Autosegmental Studies in Bantu Tone*. Foris Publications, Dordrecht, The Netherlands.

James Rogers. 1998. *A Descriptive Approach to Language-Theoretic Complexity*. (Monograph.) Studies in Logic, Language, and Information. CSLI Publications.

James Rogers and Dakotah Lambert. 2019. Extracting Subregular constraints from Regular stringsets. *Journal of Language Modelling*, 7(2):143–176.

Edward Sapir and Harry Hoijer. 1967. *The Phonology and Morphology of the Navaho Language*, volume 50 of *Linguistics*. University of California Press, Berkeley, California.

Imre Simon. 1975. Piecewise testable events. In Helmut Brakhage, editor, *Automata Theory and Formal Languages*, volume 33 of *Lecture Notes in Computer Science*, pages 214–222. Springer-Verlag, Berlin.

Howard Straubing. 1985. Finite semigroup varieties of the form $V * D$. *Journal of Pure and Applied Algebra*, 36:53–94.

Борис Авраамович Трахтенброт. 1961. Конечные Автоматы и Логика Одноместных Предикатов. *Доклады Академии наук СССР*, 140(2):326–329.

A Python Implementation

```
def lt():
    def R(F, sym):
        if sym is None: return F
        return F | frozenset([sym])
    return R, frozenset()
def ltt(t):
    def R(F, sym):
        if sym is None: return F
        d = dict(F)
        d[sym] = min(t, d.get(sym, 0) + 1)
        return frozenset(d.items())
    return R, frozenset()
def dd1(j):
    def R(F, sym):
        if sym is None: return F
        return ( F
               | frozenset([(sym,)])
               | frozenset((s + (sym,))[-j:] for s in F))
    return R, frozenset([tuple()])

def blockifier(k, state, sym=None):
    if sym is None:
        if len(state) < k: return '^' + state + '$'
        if k == 0: return None
        return state + '$'
    if len(state) == k: return state + sym
    if len(state) == k - 1: return '^' + state + sym
    return None

def factorize(k, R, alpha, aut, *args, approximate=False):
    if k < 1: raise ValueError('k must be at least one')
    stack = [(aut['q0'], '', alpha)]
    visited, accepted, rejected = set(), set(), set()
    while stack:
        q, islq, F = stack.pop()
        visited.add((q, islq, F))
        nexts = set()
        for letter,qn in aut['delta'][q].items():
            nexts.add((aut['delta'][q][letter],
                       (islq + letter)[-(k - 1):][:k - 1],
                       R(F, blockifier(k - 1, islq, letter))))
        m = R(F, blockifier(k - 1, islq))
        if q in aut['Qf']:
            if m in rejected and not approximate: return None
            accepted.add(m)
        else:
            if m in accepted and not approximate: return None
            rejected.add(m)
        stack.extend(nexts - visited)
    return accepted
```

Here are some examples. Output is omitted, but each runs nearly instantly.

```
karanga_shona = {
    'delta': {
        0 : {'H':  0, 'L':  0}, # rejecting sink
        1 : {'H':  2, 'L':  9},
        2 : {'H':  3, 'L':  8},
        3 : {'H':  4, 'L':  7},
        4 : {'H':  0, 'L':  5},
        5 : {'H':  6, 'L':  5},
        6 : {'H':  0, 'L':  0},
        7 : {'H':  6, 'L':  0},
        8 : {'H':  6, 'L':  0},
        9 : {'H': 10, 'L':  0},
        10: {'H': 11, 'L':  6},
        11: {'H':  0, 'L': 12},
        12: {'H':  0, 'L': 12},
    },
    'q0': 1,
    'Qf': set([2, 6, 8, 9, 10, 12])
}
print(*factorize(4, *lt(), karanga_shona), sep='\n')
```

```
culminativity = {
    'delta': {
        0 : {'p':  0, 'u':  0}, # rejecting sink
        1 : {'p':  2, 'u':  1},
        2 : {'p':  0, 'u':  2},
    },
    'q0': 1,
    'Qf': set([1,2])
}
print(*factorize(2, *ltt(2), culminativity), sep='\n')
```

```
murik = { # from StressTyp2
    'delta': { # alphabet: w0.s0=l, w0.s2=L, w1.s2=H
        0: { 'l':0, 'L':0, 'H':0 }, # rejecting sink
        1: { 'l':3, 'L':2, 'H':2 },
        2: { 'l':2, 'L':0, 'H':0 },
        3: { 'l':3, 'L':0, 'H':2 },
    },
    'q0': 1,
    'Qf': set([2])
}
print(*factorize(1, *dd1(2), murik), sep='\n')
```

Examining the Limits of Word2Vec with Toki Pona

Daniel Zhenhan Huang
Georgia Institute of Technology
Atlanta, GA
dzh@gatech.edu

Hongchen Wu
Georgia Institute of Technology
Atlanta, GA
hwu480@gatech.edu

Abstract

Word2Vec's effectiveness at generating semantic embeddings has been widely validated, yet it has been tested almost exclusively on languages with large vocabulary inventories. This study examines whether Word2Vec can successfully capture semantic relationships within an extremely reduced vocabulary using data from Toki Pona, a constructed language with approximately 130 words. We sourced 1.4 million sentences (7.95 million tokens) from the Toki Pona community for training. Approximately 23% of sentences in the corpus contain non-Toki Pona tokens such as named entities, loanwords, and neologisms. To investigate whether this linguistic noise enhances or hinders performance—a topic rarely addressed in word embedding literature—we trained two distinct models: one retaining these incidental tokens and another filtering them out completely. Evaluation was conducted using quantitative methods measuring word proximity to semantic category centroids, automated silhouette scores via agglomerative clustering, and qualitative analysis utilizing representational similarity matrices compared against English. The results indicate that while sparse, non-core tokens do not affect the relative structure of the learned embeddings, they actually draw similar words closer together in the vector space. Importantly, Word2Vec's effectiveness depends more on distributional patterns than lexicon size even at this extreme lower bound.

1 Introduction

Word embeddings are a foundational tool in natural language processing, capturing words' semantic relations from their distributional patterns in text (Mikolov et al., 2013). Their effectiveness has been repeatedly validated, but almost entirely on natural languages whose vocabulary size ranges from tens of thousands to hundreds of thousands of words. What happens when we push Word2Vec to an extreme edge case, significantly reducing its vocabulary's size?

Toki Pona is a lexically minimal constructed language created by Sonja Lang in 2001, designed for explicit and deliberate expression with a vocabulary of approximately 130 core words (Lang, 2014; Asi, 2025). As such, words in Toki Pona are often broad and polysemous: *lipu*[1] refers to paper or flexible sheets, but also extends abstractly to documents like websites and blogs; *tomo* covers physical enclosures like rooms, but also vehicles. The minimization of content words creates an environment radically different from those for which Word2Vec was designed.

Distributional models like Word2Vec rely on co-occurrence profiles between pairs of words to distinguish words' meanings. With a vocabulary of only 130 words, most words will co-occur with others at a much higher frequency, potentially making these profiles less discriminative. This raises a question about Word2Vec's efficacy: does an effective model require a large vocabulary with fine lexical distinctions to recover meaning, or can meaningful structure emerge even in Toki Pona's minimalist vocabulary?

A complicating factor is that approximately 23% of sentences in the gathered corpus contain non-Toki Pona tokens—proper names, loanwords, and neologisms that fall outside the language's 130-word core vocabulary. While these tokens reflect natural use, their impact on embedding quality is unclear; they may provide additional context that improves the quality of semantic relationships, or they may obscure the structure of the core vocabulary. As such, we train two Word2Vec models on a corpus of 1.4 million Toki Pona sentences with and

[1] In Toki Pona orthography, all core vocabulary is written in lowercase regardless of sentence position; only proper names, which are adapted to Toki Pona phonotactics and preceded by a noun, are capitalized. We follow this convention when citing Toki Pona forms. The English name of the language is capitalized per English convention.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 37–46
July 3-4, 2026 ©2026 Association for Computational Linguistics

without these non-Toki Pona tokens and evaluate the resulting embeddings to understand what the language's nature reveals about the capabilities of word embedding models.

We further compare the models against an independently trained English model using representational similarity analysis. Consistent patterns between semantic categories across languages would provide convergent validity and indicate that the models learn real semantic structure, not just artifacts of the corpus or of our category definitions.

2 Methodology

2.1 Data

Despite its minimalist vocabulary, Toki Pona enjoys a speaking community of thousands (van der Meulen, 2021). Because the language gained popularity primarily through the internet, the vast majority of its use is online. Its largest speaking community, *ma pona pi toki pona* (lit. 'a good place for Toki Pona'), is a Discord server with over 18,000 members (*ma pona pi toki pona*, 2025).

The corpus used in this paper consists of conversational data collected from public channels in this community, spanning from early 2016 through November 2025. After filtering to retain only Toki Pona sentences with Danielson's (2025) *sona toki* library and removing single-token sentences, the corpus contains a total of 1.42 million sentences with 7.95 million tokens.

While Danielson's (2025) method reliably determines whether a sentence can be considered to be in Toki Pona using heuristics such as Toki Pona word frequency and phonotactic conformity, many sentences still contain incidental non-Toki Pona words such as named entities (e.g., *Sonja*, the language's creator, and *Inli*, the Toki Pona adaptation of *English*) and other loanwords. These non-Toki Pona tokens may serve as "noise", obscuring the semantic relationships between core vocabulary items, or they may provide additional context that ultimately strengthens the embeddings' effectiveness. To better understand the impact of these additional tokens on the semantic structure of the model, a subset of the corpus is created with all sentences containing non-Toki Pona tokens removed, totaling 1.10 million sentences and 5.90 million tokens. We refer to these as the **Full Corpus model** and **Pure Toki Pona model** respectively.

We categorized token types present in the Full Corpus but not the Pure Toki Pona corpus with at least 50 occurrences, as shown in Figure 1. The most frequent omissions were those of *named entities*, i.e., proper names adapted to fit Toki Pona's phonotactics like *Sonja* and *Inli*, making up over 110,000 tokens. Nonstandard words (e.g., *lanpan* 'to take, steal') and neologisms (e.g., *eliki* 'adversity; bittersweet') were the second largest category by usage, with over 55,000 occurring tokens. Unadapted words, i.e., direct loans such as *Discord* from non-Toki Pona languages without adaptation, appeared less frequently than nonstandard words but represented a larger number of unique types, suggesting that code-switching, while infrequent, spans a wide variety of terms. In the following sections, we will examine whether these words provide additional useful information or simply introduce "noise" into the Full Corpus model when generating word embeddings.

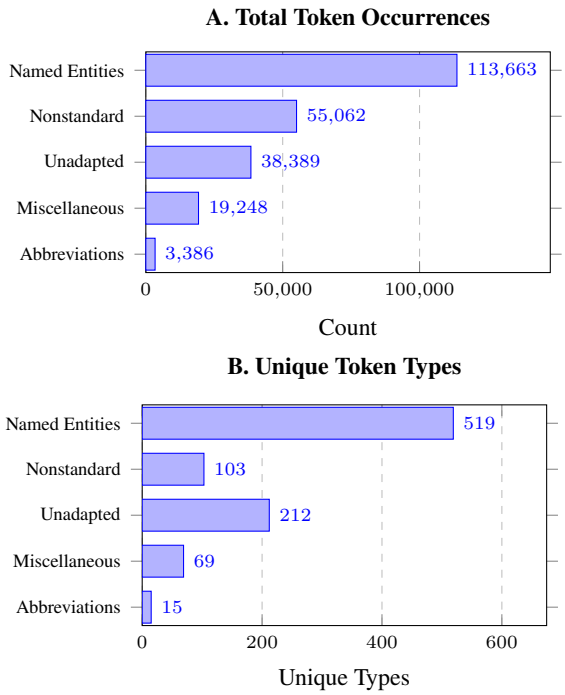


Figure 1: Distribution of non-Toki Pona tokens. Named entities adapted to Toki Pona phonotactics make up the largest fraction of the volume of removed words. Though nonstandard words make up a large volume of tokens, unadapted loanwords contribute a higher number of unique types.

2.2 Training

Two models are trained from scratch—one on the Full Corpus and the other on the Pure Toki Pona corpus.

The models are trained with skip-gram with a

window size of 5 over 10 epochs. Skip-gram was chosen for its stronger performance in semantic tasks (Mikolov et al., 2013). A major consideration during training is the vector dimensionality hyperparameter; typical models for natural languages use between 100 and 300, but these are for vocabularies of hundreds of thousands of words. Yin and Shen (2018) develop methods to determine the optimal dimensionality based on a lexicon size, provided corpus, and training algorithm; this was used on the corpus and vocabulary, yielding an optimal size of 24 dimensions. Trained models (see Appendix C) are available along with the publication of this paper.

2.3 Model Evaluation

Several evaluation metrics that work on languages with large vocabularies are incompatible with the nature of Toki Pona; for instance, evaluating with analogy performance is impractical due to the fact that Toki Pona's words generally lack parallels. As such, the models are evaluated on two metrics: category centroid proximity with a set of ground truth categories and automated evaluation with agglomerative clustering.

2.3.1 Category Centroids

Transferring existing benchmarks such as SimLex-999 (Hill et al., 2015), WordSim-353 (Finkelstein et al., 2001) to Toki Pona is infeasible, due to both the small size of its lexicon and the broad semantic scope of its vocabulary compared to other languages. Instead, this study uses a variation on Tsvetkov et al.'s (2015) approach to English model evaluation, creating categories based on WordNet supersenses (Miller, 1995). While WordNet's original categories do provide semantic groupings for nouns and verbs, adjective supersenses are grouped by syntactic rather than semantic categories; therefore, we based our categories on Tsvetkov et al.'s (2014) adjective supersenses in addition to the original noun and verb senses.

Applying English WordNet supersenses directly to Toki Pona may introduce an Anglocentric bias in evaluation (Bender, 2011). To create an effective ground truth for evaluation, categories were adapted to reflect Toki Pona's semantic structure. For instance, because of Toki Pona's small vocabulary, words frequently change parts of speech depending on their syntactic position (Lang, 2014). As such, distinctions do not exist between categories like NOUN.MOTION and VERB.MOTION, and such categories were merged where applicable. Function words like syntactic particles (e.g., *li*, *e*, *pi*) and pronouns (e.g., *mi*, *sina*) were excluded from this analysis to focus on semantic relations. In total, 27 categories were developed over 108 content words, with 125 word-category assignments. For the complete mapping, see Appendix A.

For each category C, a centroid vector μ_C is calculated by averaging the normalized word vectors v_w for all words w in that category.

$$\mu_C = \frac{1}{|C|} \sum_{w \in C} v_w \tag{1}$$

Because words in Toki Pona may belong to multiple domains, we avoid a "winner-take-all" measurement of the closest category. Instead, to determine if a word has been successfully categorized, we check whether its cosine similarity to each of its categories' centroids is above a dynamic threshold for each particular assignment. The threshold is determined by the particular word's mean similarity across all categories plus one standard deviation ($\mu + \sigma$). To validate the model's alignment with human judgment, we measure the percentage of words that are successfully retrieved, i.e. had a similarity over the dynamic threshold, by their assigned categories.

2.3.2 Unsupervised Clustering

To validate semantic relationships without human-biased semantic categories, agglomerative hierarchical clustering was used to build up clusters based on the locations of words in the vector space. Clusters were combined with the complete linkage criterion; i.e., the distance between two clusters was considered the distance between their two most dissimilar members. This was chosen to favor compact, spherical clusters, and to prevent loosely related words from forming long and incoherent groups that might arise from Toki Pona's polysemy.

Rather than specifying a number of clusters k, a hyperparameter sweep was performed over the distance threshold parameter from 0.50 to 0.90, allowing the model to determine the natural number of semantic clusters with varying levels of strictness. The resulting groupings were evaluated with their silhouette scores, selecting the distance thresholds that maximized the value.

Furthermore, a random baseline was established to verify that the resulting silhouette scores' values were statistically significant. A random normal distribution for 108 categorized words was evaluated

Metric	Full Corpus	Pure Toki Pona
Silhouette Score	0.261	0.267
Threshold	0.38	0.58

Table 1: Comparison of clustering metrics between the Full Corpus model and the Pure Toki Pona model.

with the same hierarchical clustering algorithm 30 times.

2.3.3 Cross-Linguistic Comparison

We use a method informed by Beinborn and Choenni's (2020) approach to analyze cross-linguistic drift, using a representational similarity analysis matrix to compare the overall semantic structure of Toki Pona with English. This is done by mapping the Toki Pona categories with their corresponding English translations. For English, we used the pre-trained embeddings trained on the Google News corpus (Mikolov et al., 2013), which contains 300-dimensional vectors for approximately 3 million unique tokens.

3 Results

3.1 Agglomerative Clustering

Both Toki Pona models achieved peak silhouette scores significantly above the random baseline. As shown in Table 1, the baseline had a mean silhouette score of 0.166 and a standard deviation of $\sigma = 0.0120$; the Full Corpus model's score of 0.261 and the Pure Toki Pona model's score of 0.267 are significantly higher than the baseline.

The difference in silhouette scores between the Full Corpus model and Pure Toki Pona model is small; the relative stability of the metrics between the two models indicates that the sparse non-Toki Pona data did not have a large impact on the general structure of generated word embeddings. A visualization confirms this stability; Figure 2 presents a UMAP (McInnes et al., 2018) projection of the vocabulary for both models, colored by their nearest centroid category. The arrangement of clusters and relative locations of words within those clusters indicate that the structure is largely invariant to the presence of occasional non-Toki Pona tokens. For instance, the KINSHIP cluster on the upper right of both projections remains consistently isolated from the remaining vocabulary, along with the ANIMALS and COLORS categories.

Similarly, the jump in the dynamic threshold from 0.38 (Full Corpus model) to 0.58 (Pure Toki Pona model) indicates that in the Pure Toki Pona model, words are more similar to their category centroids on average. This suggests that without the external non-Toki Pona tokens, the minimalist vocabulary of Pure Toki Pona creates a much denser, more interconnected vector space where a higher baseline similarity is required to distinguish a specific category assignment from general semantic noise.

Furthermore, qualitative inspection of the generated clusters demonstrates that both models successfully recover distinct semantic domains without supervision. In both models, ANIMALS (e.g., *soweli*, *waso*, *kala*) and COLORS (e.g., *loje*, *laso*, *jelo*) consistently formed exclusive clusters at optimal distance thresholds. These groupings emerged independently of the manually defined categories used for the centroid analysis, suggesting that the embeddings do truly capture the semantic structure of the language.

While several categories such as ANIMALS, KINSHIP, and COLORS are relatively isolated from the rest of the vocabulary, multiple clusters exhibit significant overlap, notably SHAPES, BODY, and DIRECTION. This intersection is largely due to Toki Pona's radical polysemy; words for shapes and directions often also double as words for body parts (e.g. *nena* 'bump, nose'; *linja* 'rope, string, hair'; *sinpin* 'front, face'; *selo* 'surface, skin'). As such, the embeddings reflect the inherent overlap of these domains in Toki Pona.

3.2 Category Centroids

Both Toki Pona models achieved similar retrieval rates with each successfully identifying 124 of the 125 word-category assignments. This corroborates the unsupervised clustering's demonstration that the sets of Toki Pona words are structurally equivalent. However, an analysis of words' proximity to their category's centroid revealed significant differences in density.

3.2.1 Semantic Density

The Full Corpus model had a higher mean cosine similarity to category centroids $\mu = 0.879$ compared to the Pure Toki Pona model's $\mu = 0.817$, a difference of -0.062. This suggests that non-Toki Pona tokens like names and loanwords are not required to form selected clusters but still provide valuable additional context that pulls related terms closer together in the vector space.

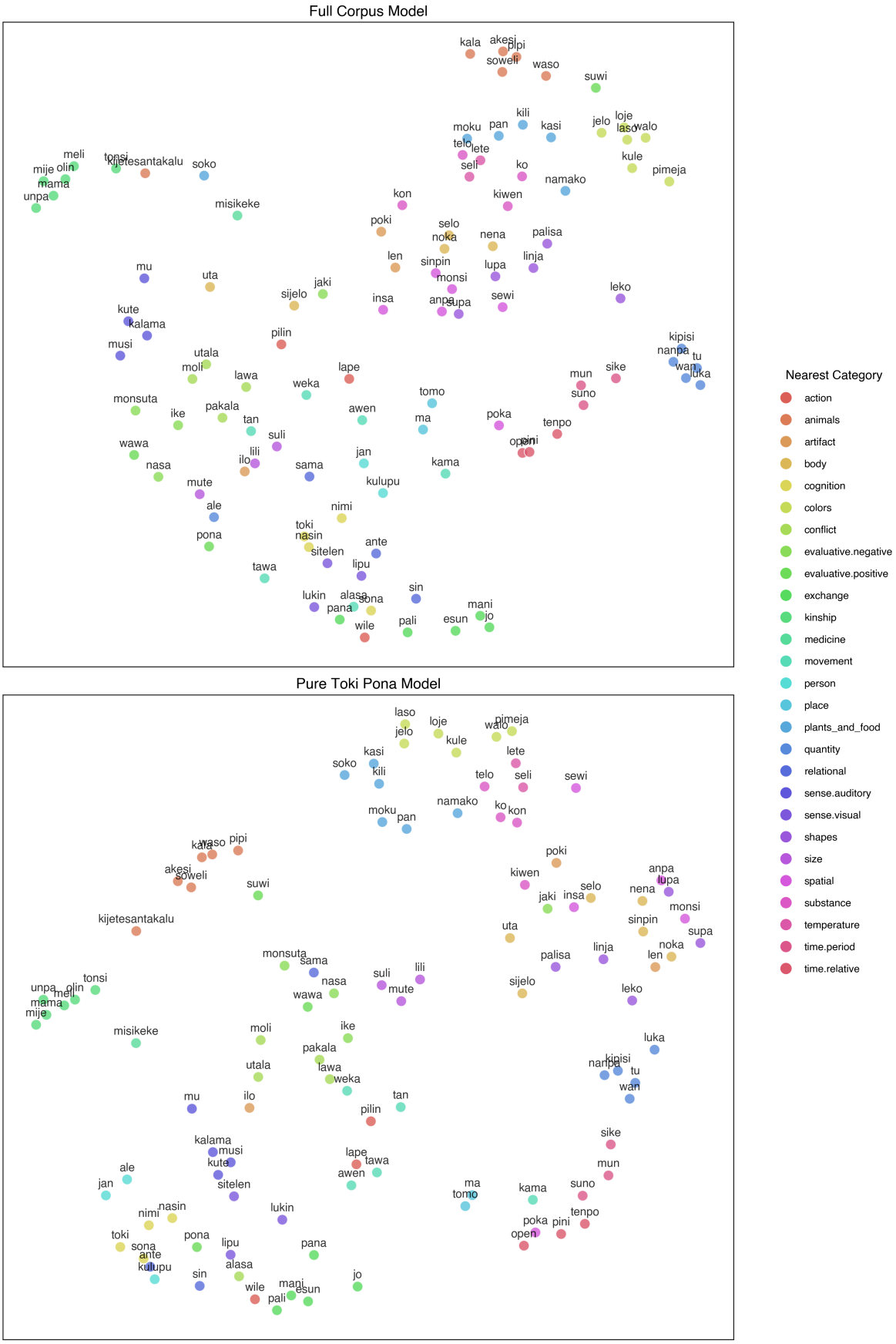


Figure 2: Full-vocabulary semantic space comparison. Models are projected by UMAP with the cosine metric and 15 nearest neighbors. Words are colored by their nearest semantic centroid.

Category	Full Corpus Score	Pure Toki Pona Score	Difference
ANIMALS	0.864	0.860	−0.004
COLORS	0.928	0.886	−0.042
SPATIAL	0.899	0.816	−0.083
EVALUATIVE.POSITIVE	0.858	0.707	−0.151

Table 2: Differences in average cosine similarity of words in specific categories to their category's centroid from the Full Corpus model to the Pure Toki Pona model.

For instance, the named entity *Jutu* 'YouTube' is closest to the vectors for *lipu* 'website', *ilo* 'tool', and *sitelen* 'image, video'. The additional data with co-occurrence of *Jutu* and these words is likely to have brought these categories closer together in the vector space, contributing to the increased cluster density.

The impact of removing non-Toki Pona tokens was not uniform across all semantic domains. Table 2 demonstrates that concrete domains like ANIMALS tend to be more stable, whereas more abstract categories like EVALUATIVE.POSITIVE are not as stable.

3.2.2 Stability of Meaning

Although the density of several categories in the Pure Toki Pona model decreased, the overall structure of the semantic groupings remained largely stable—105 of the 108 words analyzed held the same top category in both models. One of these changes was a shift between two categories—*sinpin* 'face, front' moving from SPATIAL to BODY. Another was a miscategorization of *alasa* 'search, try' as a MOVEMENT word in the Full Corpus model. The third was a miscategorization of *ale* 'all, everything' as a PERSON word in the Pure Toki Pona model. This is likely due to the frequent co-occurrence of *ale* with *jan* in the phrase *jan ale* 'everyone', as in *toki a, jan ale o* 'hello, everyone'.

3.3 Cross-Linguistic Comparison

A qualitative analysis of Figure 3 indicates that although the Toki Pona categories are more dense (as expected with the language's radical polysemy along with the lower vector dimensionality), the patterns of relative proximity between categories of words are remarkably consistent across both Toki Pona and English models. For instance, the BODY and SPATIAL categories are closer together than nearby categories, along with the ANIMALS and PLANTS AND FOOD categories. For readability, we present Figure 3 with a representative subset of eight categories here; complete matrices for all 27 categories with the same pattern are available in Appendix B.

4 Conclusion and Future Directions

This study investigated whether Word2Vec could generate meaningful embeddings within Toki Pona's highly constrained lexicon of approximately 130 words, while additionally exploring the architectural effects of non-core tokens. Our results indicate that vocabulary size does not constitute a bottleneck for Word2Vec. Even at this extreme lower bound, the model successfully encoded the semantic structure of the language's lexicon and distinguished distinct category clusters without supervision.

Furthermore, a comparison between the Full Corpus and Pure Toki Pona models demonstrates that while core semantic relationships are entirely recoverable in the pure model, the density of the resulting embeddings improves with the inclusion of occasional non-Toki Pona tokens. This suggests that these external elements—primarily proper nouns, loanwords, or instances of code-switching—act as structural bridges within the vector space, effectively refining the positioning of highly polysemous vocabulary. This validity is further supported by our cross-linguistic comparison, which demonstrates that relative distances between semantic categories remain remarkably consistent between Toki Pona and English, confirming that the embeddings capture genuine semantic structures rather than corpus artifacts.

4.1 Broader Impacts and Future Work

This work challenges the conventional assumption that fine lexical distinctions and massive vocabularies are prerequisites for effectively encoding meaning via word embeddings. By successfully training a model on a language characterized by extreme polysemy and a minimalist lexicon, we confirm that

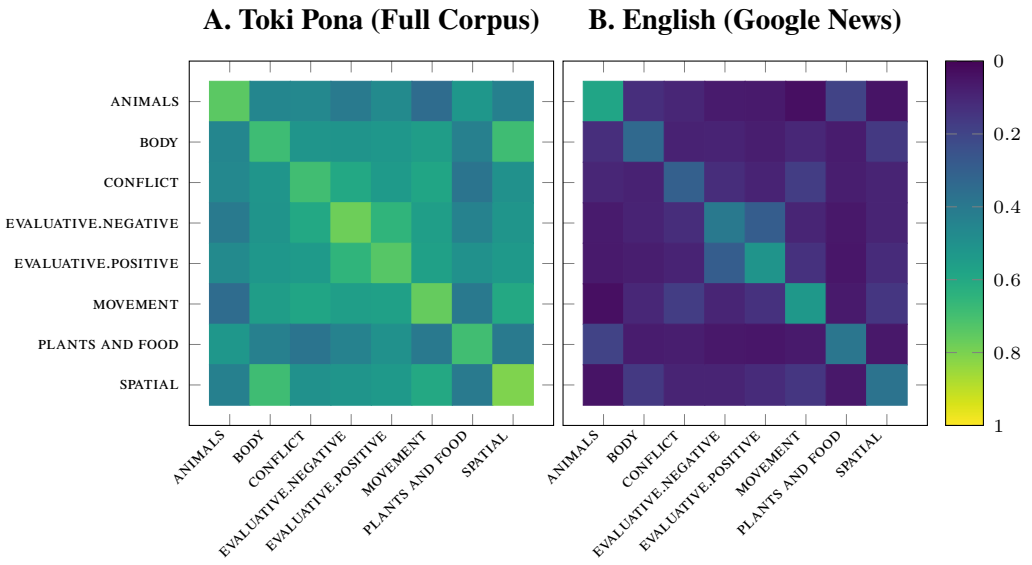


Figure 3: Representational similarity matrices with a subset of Toki Pona categories and corresponding English categories.

the distributional hypothesis holds, even at an absolute lower bound.

As a reviewer pointed out, these findings may offer crucial implications for the processing of low-resource and endangered languages in two key ways.

First, many endangered or under-resourced languages possess compact core lexicons or rely heavily on extreme polysemy and compounding (e.g., polysynthetic structures) to convey complex concepts. Our findings provide an empirical proof-of-concept that Word2Vec's architecture remains robust under these constraints, organizing semantic spaces effectively through distributional density alone.

Second, in real-world, low-resource communities, speakers frequently engage in code-switching or integrate loanwords from dominant regional languages. Our finding that non-core tokens act as a bridge to tighten the vector space without disrupting its underlying structure suggests that NLP pipelines for low-resource languages do not require perfectly curated, "pure" corpora. Instead, the inclusion of code-switched data and lexical borrowings can actively accelerate semantic alignment and stabilize sparse embeddings.

Naturally, these broader computational impacts require empirical validation in natural languages. Future work will focus on applying these insights to natural low-resource, endangered language datasets, specifically investigating how code-switching densities and varying degrees of polysemy affect embedding stability in endangered language documentation.

5 Acknowledgments

This study was supported by the President's Undergraduate Research Travel Award and Professional Development Fund (School of Modern Languages) at the Georgia Institute of Technology, as well as the SCiL 2026 Student Subsidy that helped fund conference travel. The authors are grateful to the reviewers for their constructive feedback and suggestions.

References

kala Asi. 2025. 2025 word survey results (n = 971). *lipu Linku* community survey.

Lisa Beinborn and Rochelle Choenni. 2020. Semantic drift in multilingual representations. *Computational Linguistics*, 46(3):571–603.

Emily M. Bender. 2011. On achieving and evaluating language-independence in NLP. *Linguistic Issues in Language Technology*, 6.

Gregory Danielson, III. 2025. sona toki. GitHub repository.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. volume 20, pages 406–414.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Sonja Lang. 2014. *Toki Pona: The Language of Good*. Tawhid.

ma pona pi toki pona. 2025. ma pona pi toki pona. Online. Discord server.

Leland McInnes, John Healy, and James Melville. 2018. UMAP: Uniform manifold approximation and projection for dimension reduction. *Preprint*, arXiv:1802.03426.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Preprint*, arXiv:1301.3781.

George A. Miller. 1995. WordNet: a lexical database for English. *Commun. ACM*, 38(11):39–41.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. 2015. Evaluation of word vector representations by subspace alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2049–2054, Lisbon, Portugal. Association for Computational Linguistics.

Yulia Tsvetkov, Nathan Schneider, Dirk Hovy, Archna Bhatia, Manaal Faruqui, and Chris Dyer. 2014. Augmenting English adjective senses with supersenses. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4359–4365, Reykjavik, Iceland. European Language Resources Association (ELRA).

Spencer van der Meulen. 2021. Request for new language code element in ISO 639-3.

Zi Yin and Yuanyuan Shen. 2018. On the dimensionality of word embedding. In *Advances in Neural Information Processing Systems*, volume 31, pages 895–906.

A Toki Pona and Corresponding English Categories

The full mapping between categories and Toki Pona words used for the centroid distance analysis is detailed in Table 3 along with the English words used for the cross-linguistic comparison.

B Full Representational Similarity Matrices

Figure 4 presents the complete representational similarity matrices for all 27 categories evaluated.

C Code

All code, trained models, and evaluation data are available at `https://doi.org/10.5281/zenodo.20297806`.

Category	Toki Pona	English
ACTION	lape, musi, olin, pilin, wile	desire, feel, play, sleep, want
ANIMALS	akesi, kala, kijetesantakalu, pipi, soweli, waso	animal, bird, fish, insect, mammal, reptile
ARTIFACT	ilo, len, poki	box, cloth, container, machine, tool
BODY	lawa, luka, monsi, nena, noka, poka, selo, sijelo, sinpin, uta	arm, back, body, face, head, leg, mouth, nose, skin
COGNITION	lipu, nasin, nimi, sitelen, sona, toki	book, image, knowledge, language, method, way, word
COLORS	jelo, kule, laso, loje, pimeja, walo	black, blue, green, red, white, yellow
CONFLICT	alasa, moli, monsuta, pakala, utala	break, fear, fight, hunt, kill, monster, war
EVALUATIVE.NEGATIVE	ike, jaki, monsuta, nasa	bad, gross, odd, scary, weak
EVALUATIVE.POSITIVE	pona, suli, suwi, wawa	awesome, cute, good, important
EXCHANGE	esun, jo, mani, pali, pana	buy, give, make, money, sell, work
KINSHIP	jan, kulupu, mama, meli, mije, olin, tonsi, unpa	brother, love, man, parent, sex, sister, woman
MEDICINE	misikeke	cure, doctor, medicine
MOVEMENT	awen, kama, tan, tawa, weka	come, go, leave, move, stay, wait
PERSON	jan, kulupu	community, group, human, person
PLACE	ma, tomo	country, house, land, room
PLANTS AND FOOD	kasi, kili, moku, namako, pan, soko, suwi	bread, food, fruit, mushroom, plant, sugar
QUANTITY	ale, kipisi, luka, mute, nanpa, tu, wan	all, five, many, number, one, part, two
RELATIONAL	ante, sama, sin	different, new, other, same
SENSE.AUDITORY	kalama, kute, mu	hear, listen, noise, sound
SENSE.VISUAL	lipu, lukin, sitelen	image, look, picture, see
SHAPES	leko, linja, lupa, nena, palisa, sike, supa	block, bump, circle, hole, line, rod, square, surface
SIZE	lili, mute, suli	big, long, short, small
SPATIAL	anpa, insa, monsi, poka, sewi, sinpin	back, bottom, center, front, inside, side, top
SUBSTANCE	kiwen, ko, kon, telo	air, gas, metal, powder, stone, water
TEMPERATURE	lete, seli	cold, cool, hot, warm
TIME.PERIOD	mun, sike, suno, tenpo	day, era, month, year
TIME.RELATIVE	kama, open, pini, tenpo	finish, future, past, start, time

Table 3: Category assignments for Toki Pona and English words.

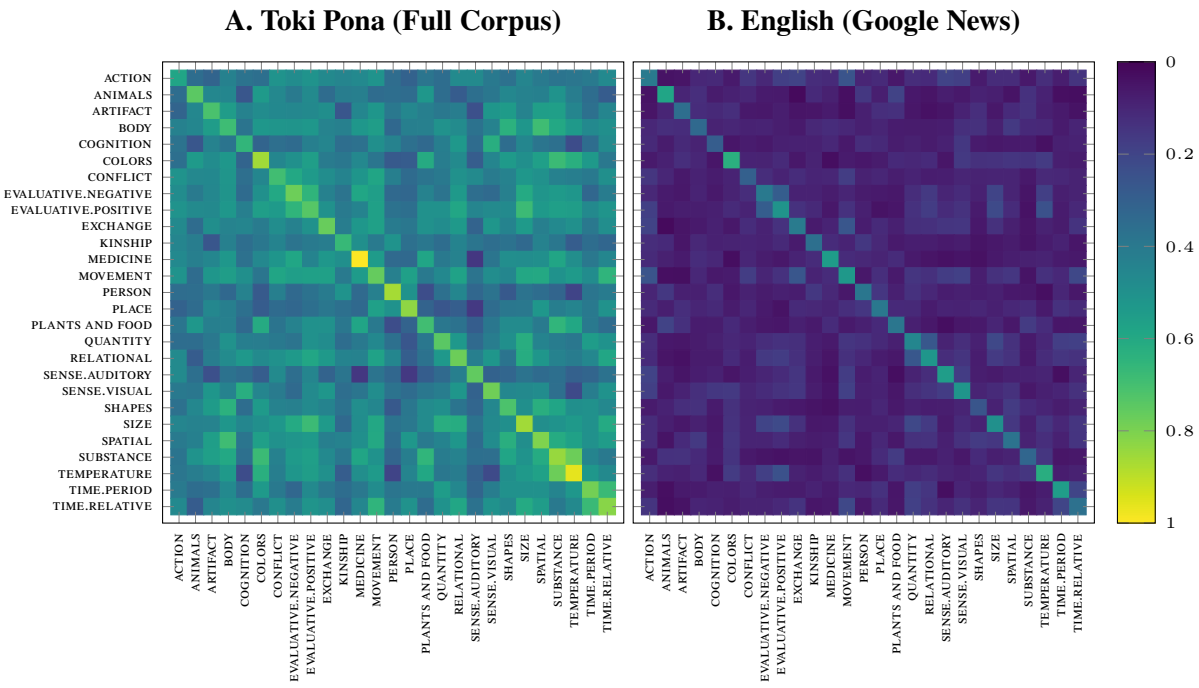


Figure 4: Complete representational similarity matrices comparing the semantic structure of all 27 defined categories in Toki Pona against English.

Learning Latent Representations with Progressive Hypothesis Space Expansion

Jonathan Charles Paramore
University of California, Santa Cruz
jcparamo@ucsc.edu

Abstract

This paper introduces a learning model to address the computational challenges arising from including highly abstract underlying representations (URs) in morphophonemic learning. The proposed learner structures the UR hypothesis space by disparity distance and considers potential URs in batches, beginning with fully concrete URs, only expanding the UR candidate space if the current set of UR candidates fails to meet a predetermined likelihood threshold. Overall, the learner inherently restricts abstraction to cases where introducing it demonstrably improves likelihood, while avoiding issues associated with the exhaustive search of an unbounded hypothesis space. Applied to Pakistani Punjabi vowel nasality, the model is shown to successfully acquire abstract URs for phonological patterns that parallel learners fail to capture.

1 Introduction

Classic debates concerning the degree to which an underlying representation (UR) and its associated surface representations (SRs) may differ (e.g., Kenstowicz and Kisseberth 1977) have reemerged in recent years. This resurgent interest owes primarily to the difficulty that increased abstraction presents for phonological learnability. On the one hand, some phonological patterns are difficult to analyze without positing highly abstract URs (e.g., Klamath O'Hara 2017, Pakistani Punjabi Paramore 2025, and Sevillian Spanish Gilbert 2023). In these cases, proposed URs are considered highly abstract in that they differ in one or more feature specifications from all of their corresponding surface exponents.

Nevertheless, despite the analytical motivations for assuming abstract URs, recent work on phonological learnability has raised serious concerns about the viability of such high degrees of UR abstraction. First, if multiple URs successfully model the observed data, what mechanism ensures that the learner converges on the minimally abstract UR? That is, how is UR abstraction permitted only to the extent necessary (Prince and Smolensky, 1993/2004)? Second, and perhaps more pressingly, if UR abstraction is unconstrained, the space of possible URs grows without bound, introducing a potentially intractable search space problem (Jarosz, 2019; Wang and Hayes, 2025).

While a number of learning algorithms have been proposed that minimize UR abstraction within Maximum Entropy frameworks (O'Hara, 2017; Paramore, 2025), the issue of search space size brought on by permitting increased UR abstraction remains largely unresolved. To date, most learning models have simply stipulated a finite set of potential URs for the learner to consider (O'Hara, 2017; Paramore, 2025). Wang and Hayes (2025), on the other hand, demonstrate that at certain levels of UR abstraction, the space of potential URs grows too large, preventing the learner from converging on a correct solution. Given these results, Wang and Hayes (2025) propose that the boundary on permissible UR abstraction should be defined by learnability itself. This conclusion places learnability-based limits on abstraction in direct tension with patterns in languages like those discussed above that appear to require highly abstract URs.

This paper introduces a novel learning algorithm—**the UR Progressive Hypothesis Space Expansion (UR PHaSE) learner**—which aims to reconcile these competing demands. The UR PHaSE learner solves both of the above issues posed by UR abstraction without imposing an explicit upper bound on the degree of UR abstraction. The central insight builds on work by Tesar (2014, 2016), which structures the UR search space so that, rather than evaluating an infinite set of potential URs in parallel, the learner partitions the space into ordered subsets and considers candidates incrementally based on disparity count. Beginning with

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 47–56
July 3-4, 2026 ©2026 Association for Computational Linguistics

i.	[sɑ]	'breath'	ii.	[sɑ̃ʊ̃ɑ̃]	'breaths'
iii.	[sɑ̃]	'I was'	iv.	[sɑ̃ɳ]	'grindstone'
v.	[sijɑ̃ɳ]	'recognition'	vi.	[dʒəʊɑ̃ɳ]	'a youth'

Table 1: Forms illustrating contrastive vowel nasality and nasal harmony in Punjabi.

fully concrete URs, the learner only expands the space to include increasingly abstract URs when it fails to find an adequate learning solution at a lower level of abstraction. Moreover, a key novelty of the UR PHaSE learner is that expansion of the UR hypothesis space is *targeted*: the learner uses information from constraint weights and violations in observed forms to generate only those candidate URs that are potentially informative for learning. In this way, the hypothesis space remains manageable as the learner iteratively considers increasingly abstract UR candidates.

Using a vowel nasality pattern from Pakistani Punjabi (Paramore, 2025; Paramore and Bennett, 2025), I show that an appropriately structured hypothesis space of potential URs coupled with a serial search algorithm renders the required level of UR abstraction learnable without needing to provide external stipulations on the UR candidate set.

2 Pakistani Punjabi Nasality

Punjabi is an Indo-Aryan language spoken by approximately 110 million people. Paramore and Bennett (2025) describe a vowel nasality pattern in Pakistani Punjabi that requires vowels occurring before tautomorphemic nasal consonants (pre-N vowels) to be represented with abstract URs containing a nasality feature that never surfaces faithfully.

As shown by examples (i) and (iii) in Table 1, vowel nasality is contrastive in Punjabi. Importantly, two phonological processes shape the realization of nasality across words. First, all pre-N vowels are categorically nasalized, as shown in Table 1iv-vi. In addition, contrastive nasal vowels trigger regressive nasal harmony, such that nasality spreads to all glides and vowels, while other consonants block further propagation (Table 1i-ii). Crucially, however, while Paramore and Bennett (2025) show that pre-N vowels are phonetically indistinguishable from contrastive nasal vowels, they also demonstrate that pre-N vowels do not trigger nasal harmony, as shown by the examples in Table 1v-vi.

To account for the distinct behavior of pre-N vowels and contrastive nasal vowels in their induction of nasal harmony, Paramore and Bennett (2025) analyze pre-N vowels as underlying oral, undergoing local categorical nasalization in the context of a following nasal consonant (I refer the interested reader to their paper for a more in-depth defense for analyzing pre-N vowels as abstract). Given this analysis, explaining the disparate behavior of pre-N and contrastive nasal vowels is straightforward: underlying nasal vowels trigger harmony, while derived nasal pre-N vowels do not.

Under this analysis, pre-N vowels are assigned abstract URs, oral underlyingly in order to block harmony triggering, but categorically nasal on the surface: /sijɑɳ/→[sijɑ̃ɳ], *[sĩjɑ̃ɳ]. The level of abstraction posited for these pre-N vowels exceeds what Wang and Hayes (2025) find to be learnable using a learning approach that considers all UR candidates in parallel. Specifically, they demonstrate that permitting URs containing a feature that never surfaces in any allomorph as a viable UR candidate expands the hypothesis space beyond what can be effectively searched, resulting in learning failures (Wang and Hayes 2025, p.30-31).

3 The UR PHaSE Learner

The UR PHaSE Learner integrates Maximum Entropy (MaxEnt) learning (Hayes and Wilson, 2008) with expectation-maximization (EM) learning (Jarosz, 2006a, 2015; Wang and Hayes, 2025) to jointly identify optimal URs for all morphemes in a given dataset and the constraint weights used in UR→SR mappings. Learning proceeds in four steps: (1) phonotactic learning to establish a best initial guess of constraint weights, (2) n-disparity UR expectation maximization learning, in which candidate URs with up to n disparities are evaluated; (3) Likelihood Threshold evaluation, which assesses whether the n-disparity UR candidate sets from step (2) sufficiently explain the observed data, and (4) generation of n+1 disparity UR candidates, which expands the UR candidate space when the current UR candidates fail the likelihood threshold evaluation in step (3). Steps (2)-(4) are repeated until the Likelihood Threshold is satisfied for all morphemes.

3.1 Step (1): Phonotactic Learning

The UR PHaSE learner first learns constraint weights to model phonotactic patterns before proceeding to consider URs for individual morphemes. Phonotactic learning here follows the standard

error-driven learning approach to phonotactic learning undertaken by MaxEnt learning models, similar to Hayes and Wilson (2008). The learner takes as input a list of surface forms and predefined markedness constraints initialized at a neutral weight (e.g., set at 50 with constraint bounds between [0, 100]). Using gradient descent, the learner adjusts constraint weight values to minimize a loss function made up of two components: the negative log likelihood of the observed data and an L2 Gaussian prior (cf., O'Hara 2017) that favors high-weighted markedness constraints, thus increasing grammar restrictiveness. The output of the phonotactic stage is an array of optimized constraint weights that capture the surface phonotactic patterns in the data while maximally satisfying the prior's preference for a restrictive grammar.

3.2 Step (2): n-disparity UR EM Learning

After the phonotactic stage, learning progresses to a morphologically-aware stage in which the UR PHaSE learner jointly acquires a probability distribution over the set of possible URs and the optimal constraint weights that maximize the likelihood of the surface data. Expectation maximization (EM) is employed to acquire UR probability distributions and constraint weights (Dempster et al., 1977). Jarosz (2006a,b, 2009, 2010, 2015) demonstrates that EM learning within a probabilistic OT framework can successfully learn a broad range of hidden linguistic structure, including URs, and the EM learning principles developed in that work provide the inspiration for the approach taken here (see also Wang and Hayes 2025).

Morphologically-aware learning involves the simultaneous acquisition of URs for individual morphemes and the constraint weights that map those URs to their observed surface realizations (SRs) across contexts. This creates a classic chicken-and-egg problem. In order to correctly identify the UR for a morpheme that maximizes the likelihood of its SRs, the constraint weights governing the UR→SR mappings must already be known. At the same time, in order to arrive at the optimal constraint weights that maximize the likelihood of a UR→SR mapping, it is also crucial to know a morpheme's UR. The EM algorithm provides a solution to this issue by breaking the learning problem into two iterated steps.

3.2.1 Expectation Step (E-step)

The goal of the E-step is to update prior beliefs about a hypothesis given new evidence. For an individual morpheme, the initial hypothesis of the learner is that the probability of candidate URs is equivalent across the entire set for that morpheme. The initial new evidence is comprised of the combination of constraint weights learned during the phonotactic stage and the observed SRs of the morpheme.

Under these conditions, the learner computes posterior probabilities over the space of candidate URs for each morpheme, updating the probability that each UR is the correct UR for a given SR, using Bayes' Theorem in equation 1.

$$\underbrace{\mathbb{P}(UR_i \mid SR_j)}_{Posterior} = \frac{\overbrace{\mathbb{P}(SR_j \mid UR_i)}^{Likelihood}\ \overbrace{\mathbb{P}(UR_i)}^{Prior}}{\underbrace{\sum_{UR_k \in S} \mathbb{P}(SR_j \mid UR_k)\ \mathbb{P}(UR_k)}_{Marginal}} \quad (1)$$

Bayes' theorem provides a principled mechanism for updating the learner's uncertainty over which UR is the correct representation for a given morpheme. With a set of observed SRs and fixed constraint weights, the posterior probability of a candidate UR can be computed by multiplying the joint likelihood of observing all SRs of a morpheme given that UR by the prior probability assigned to the UR, and then normalizing by the marginal probability of the surface forms across all candidate URs for the morpheme. The UR's prior probability is initially uniform across candidate URs before any learning has taken place and is re-estimated in the M-step as the average posterior responsibility assigned to that UR across the SRs of the morpheme.

Crucially, the size of a morpheme's UR candidate space is what typically results in failures during UR learning (Wang and Hayes, 2025, p.33-35). As increasingly abstract URs are permitted, the hypothesis space size explodes, making traversal across the candidate set costly and leading to learning failures (Jarosz, 2019, p.78-79).

One of the main insights of the UR PHaSE learner is that the UR hypothesis space is structured based on the number of disparities in a UR→SR mapping for each morpheme (cf., Tesar 2014, 2016). This structure is illustrated by the lattice for a hypothetical morpheme with surface forms [i] and [ĩ] and possible disparities for the features [FRONT], [HIGH], and [RD] in Figure 1.

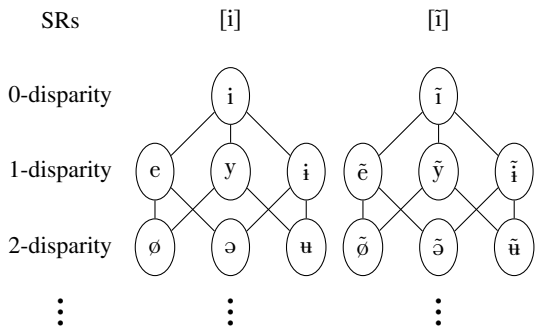


Figure 1: Lattice for a structured search space of candidate URs for a single morpheme realized contextually as [i] and [ĩ]. Visualized disparities only include the features [HIGH], [RD], and [FRONT].

As Tesar (2014, p.21) defines it, a featural disparity in a UR→SR mapping occurs when corresponding input-output segments differ in their value for a feature.[1]

The position of a UR in a morpheme's UR hypothesis space is determined by the disparity distance between that UR and the closest SR. URs that are realized faithfully in at least one allomorph count as 0-disparity URs and reside at the top of the UR space. 1-disparity URs, on the other hand, can be reached by flipping exactly one feature value in one segment for at least one of the SRs. 2-disparity URs are those that can be reached by flipping exactly two feature values—either in one segment or across two segments—for at least one of the SRs. URs with more than two disparities (not depicted in Figure 1) are positioned on the appropriate disparity level.

Importantly, while the feature flips resulting in the level 2 candidate URs in Figure 1 are implemented on a single segment, this does not have to be the case. If a morpheme is constructed of multiple segments, there must be exactly two combined feature flips across all segments for a UR to be considered a 2-disparity UR.

Underlying the motivation to structure the UR candidate space based on disparities is the assumption that learners inherently disprefer differences in UR→SR mappings (Prince and Smolensky, 1993/2004). That is, the learner assumes morphemes will be represented concretely unless evidence suggests otherwise. The framework of the UR PHaSE learner incorporates this assumption into its algorithm by considering potential UR candidates in batches, beginning with all 0-disparity candidates and only considering additional disparity candidates if the likelihood of the data remains low using URs with the current number of permitted disparities.

Note that nothing constrains the number of possible disparities a candidate UR can take, which means the entire hypothesis space will be quite large for most morphemes. Nevertheless, the UR PHaSE learner's algorithm takes advantage of the structure of the space to avoid searching it exhaustively.

3.2.2 Maximization Step (M-step)

After defining the UR candidate space for each morpheme and calculating the posterior probability distributions of each UR candidate space, the learner moves to the M-step, seeking to maximize the likelihood of the observed data (Jarosz, 2006a,b). To do so, the learner updates the constraint weights (**w**), using gradient descent to minimize a loss function made up of the negative log likelihood of all n observed SRs (2a) and an L2 Gaussian prior (2b) that maximizes grammar restrictiveness by preferring high-weighted markedness constraints and low-weighted faithfulness constraints (O'Hara, 2017; Paramore, 2025).[2,3]

$$\text{NLL} = -\ln\left[\prod_{i=1}^{n}\left(\mathbb{P}[SR_i \mid (\mathbf{w}, \pi_{\mathbf{post}})]\right)\right] \quad (2a)$$

$$\text{L2 Prior} = \sum_{w_i \in \mathbf{w}} \frac{(w_i - c_i)^2}{\sigma_i^2} \quad (2b)$$

Once the optimal constraint weights are discovered in the M-step, the learner returns to the E-step, updating these posterior probability distributions based on the newly optimized constraint weights. Iteration between these steps continues until the learner converges on stable constraint weight values and UR probability distributions.

3.3 Step (3): Likelihood Threshold Evaluation

Convergence on a set of optimal UR distributions and constraint weights in the EM learning step

[1] I set aside disparities caused by insertion and deletion in this paper due to space constraints, but future work will incorporate them into the learner.

[2] See Hayes and Wilson (2008) for a detailed explanation of gradient *ascent*, the maximization counterpart to the gradient descent process used here.

[3] To maximize restrictiveness, c_i is set to 100 for markedness constraints and 0 for faithfulness constraints. σ_i^2 is a plasticity constant used to modulate how much deviations from ideal weights impact the value of the loss function.

is not equivalent to solving the learning problem. Rather, it means that the learner has arrived at a locally optimal solution with the current UR candidates under consideration for each morpheme, but it does not mean this solution is adequate. To determine adequacy, step (3) evaluates the likelihood of each individual SR in the observed data using the UR probabilities and constraint weights learned in step (2). If all SRs exceed a .95 likelihood threshold, the learner—satisfied that the current URs generate the observed data with sufficiently high probability—stops its search for a more optimal model. If, however, any single SR exhibits less than .95 likelihood, the learner has failed to meet the Likelihood Threshold and continues iterating through the learning pathway (cf., Belth 2026).

Establishing a Likelihood Threshold at which the learner is satisfied with its learned representations is a defining feature of the UR PHaSE learner, and is similar to how humans approach decision making. Rather than examine all possible candidates in a hypothesis space exhaustively, humans almost always settle on a solution once it meets some threshold that satisfies their expectations (Simon, 1955, 1956). Of course, it is certainly true that acceptability thresholds vary from individual to individual and decision to decision. In this sense, the .95 likelihood threshold offered here serves as a stand-in for individual variation of what a satisfactory solution may be.

3.4 Step (4): UR Candidate Expansion

If any surface form falls below the .95 threshold in step (3) using n-disparity URs, the UR PHaSE learner will expand the UR hypothesis space for all morphemes by introducing n+1-disparity UR candidates. Crucially, even at low-disparity levels, the unrestricted generation of UR candidates can cause the hypothesis space to grow too large to support effective learning.

To avoid this combinatorial explosion, the UR PHaSE learner does not generate all logically possible n+1-disparity URs. Instead, it exploits information already present in the grammar: only features associated with markedness constraints that (i) are violated by at least one surface representation and (ii) currently exhibit low (i.e., non-maximal) weights are eligible to be flipped to generate new UR candidates.

This targeted strategy equips the learner with a principled means of expanding the UR search space. Rather than indiscriminately introducing all higher-disparity candidates, the learner proposes only those URs that introduce additional UR→SR mapping candidates that make violations of these markedness constraints informative for learning. In this way, the *selective* introduction of UR candidates at a single level of increased abstraction provides the learner with precisely the evidence needed to revise its grammatical commitments, while keeping the hypothesis space tractable.

i.	[CV]	stem1	ii. [CṼ-G̃Ṽ]	stem1-suff1
iii.	[CṼ]	stem2	iv. [CṼ-G̃Ṽ]	stem2-suff1
v.	[CVGṼN]	stem3		

Table 2: Punjabi surface form categories

3.5 Iteration Until Convergence

After the UR candidate space is expanded in step (4), the UR PHaSE learner repeats the learning process in steps (2)-(4) until all morphemes exceed the Likelihood Threshold evaluation in step (3).

4 A Test Case: Punjabi Pre-N Vowels

This section demonstrates that the level of UR abstraction required for pre-N vowels in Punjabi is learnable under the architecture of the UR PHaSE learner. The model was implemented in Python. The code and data used in this section are available at github.com/joncparamore/UR_PHaSE.

Surface forms exemplified by the five word categories in Table 2 served as the input data for the learning model. The entire dataset is provided in the appendix. Six words were included from each of the five categories, resulting in a total of thirty forms. Along with the observed surface form of each word, the learner was also provided with morpheme boundary information.[4] Categories in Table 2.i-ii demonstrate the nasal harmony pattern triggered by contrastive nasal vowels, and the patterns in Table 2.iii-iv illustrate that vowel nasality is contrastive. Finally, the category in Table 2.v demonstrates that pre-N vowels are nasalized but do not trigger harmony.

4.1 Step (1): Phonotactic Learning

The ten constraints listed in Table 3, along with their definitions (provided in the appendix), were also supplied to the learner. All constraint weights were initialized at the midpoint value of 50 at the

[4] See Wang and Hayes (2025) for a method of learning morpheme boundaries during morphophonemic learning.

Constraint	Type	initial w	final w
SPRD-L[NAS]	mark.	50.00	0.13
*NASOBS	mark.	50.00	100.00
*NASG	mark.	50.00	0.47
*NASV	mark.	50.00	0.00
*VN	mark.	50.00	100.00
ID[NAS]	faith.	50.00	0.00
IDFin[NAS]	faith.	50.00	0.00
ID[NAS]/_V	contfaith.	50.00	0.00
ID[RD]	faith.	50.00	0.00
*LOWRD	mark.	50.00	100.00

Table 3: Constraint weights with their initial and optimized weights in the phonotactic stage.

outset of step (1). ID[NAS]/_V is a unique constraint in that it is a contextual faithfulness constraint (Hauser and Hughto, 2020) that penalizes changes to a segment's [NAS] feature value when directly preceding a vowel specified as [-NAS] in the *input*. As is well known, parallel versions of Optimality Theory are unable to capture many opaque phonological processes (McCarthy, 2000, 2007), including the opaque counterfeeding pattern in which pre-N vowels are nasalized on the surface but do not trigger harmony in Punjabi. To address this, contextual faithfulness constraints reference a segment's input context to explain why vowels and glides preceding pre-N vowels do not undergo nasalization.

The weights learned in the phonotactic stage are provided in the final column of Table 3 and serve as the input to the morphologically-aware learning stage that follows. From a cursory examination, it is clear that the phonotactic grammar correctly bans nasal obstruents, oral vowels before nasal consonants, and low round vowels—none of which occur in Punjabi—by maximizing the associated markedness constraints. Also, because the grammar does not consider URs during the phonotactic stage, all faithfulness constraints were reduced to zero to allow for a maximally restrictive grammar. *NASG must have a low weight during the phonotactic stage because nasal glides are phonotactically licit in Punjabi (e.g., [sãṽ̃ã] 'breaths'). Likewise, because pre-N forms do not exhibit nasal harmony (e.g., [tɑʊ̃n] 'penalty'), SPRD-L[NAS] must also be assigned a low weight in the phonotactic stage. Otherwise, the grammar would impose nasal harmony in all applicable environments, including in pre-N forms.

4.2 Step (2): 0-disparity UR EM Learning

After phonotactic learning, step (2) uses the Expectation–Maximization (EM) algorithm to search for the combination of URs and constraint weights that maximize the likelihood of the observed surface forms.

Although the learner optimizes UR probability distributions for all morphemes across the dataset, I report updates for only three representative word forms here: [sã] 'breath', [sãṽ̃ã] 'breaths', and [tɑʊ̃n] 'penalty'. Together, these forms are sufficient to diagnose whether the learner has acquired (i) nasal harmony triggered by contrastive nasal vowels and (ii) the absence of nasal harmony when vowel nasality is derived in pre-N contexts.

In this round of EM learning, the UR PHaSE learner constructs the UR candidate space for each morpheme based solely on the surface alternations it has encountered. The 0-disparity UR candidates for the morphemes 'breath' and 'penalty' (Table 4) are all identical to at least one SR of their respective morpheme.

The results of EM learning using only 0-disparity URs are given in Table 4. Note that the final posterior UR probabilities the learner converged on remain unchanged from their initial uniform distributions. Additionally, the likelihood of observing [tɑʊ̃n] 'penalty' is 1, indicating that nasal harmony is correctly not expected in pre-N forms. For the two realizations of 'breath' [sɑ]∼[sãṽ̃ã], however, the learner is unable to correctly model nasal harmony given the current UR options, so instead optimizes the likelihood as much as possible by distributing UR probabilities evenly across the two candidate URs. This results in approximately 50% likelihood of observing the correct nasality patterns of 'breath'.

The reason for the learning failure is straightforward: given only concrete forms as UR candidates, the learner cannot find constraint weights that induce nasal harmony in 'breath' while simultaneously suppressing it in the pre-N form 'penalty'.

4.3 Step (3): Likelihood Threshold Eval. for 0-disparity URs

As is already evident from the sampling of SRs in Table 4 above, the six singular surface forms with an oral vowel in the stem (like [sɑ] 'breath') and the six surface forms with those same stems in the plural (like [sãṽ̃ã] 'breaths') all fail to meet the Likelihood Threshold of .95 when using only

Constraint	Type	initial w	final w
SPRD-L[NAS]	mark.	0.13	62.92
*NASOBS	mark.	100.00	100.00
*NASG	mark.	0.47	60.24
*NASV	mark.	0.00	23.16
*VN	mark.	100.00	100.00
ID[NAS]	faith.	0.00	83.44
IDFin[NAS]	faith.	0.00	89.46
ID[NAS]/_V	contfaith.	0.00	100.00
ID[RD]	faith.	0.00	6.29
*LOWRD	mark.	100.00	100.00

Morpheme	UR candidate	Prior	Posterior
'breath'	/sɑ/	0.5	0.5
	/sɑ̃/	0.5	0.5
'penalty'	/tʌvɑ̃n/	1.0	1.0

SR	Likelihood
[sɑ]	0.50
[sɑ̃ʊ̃ɑ̃]	0.48
[tʌvɑ̃n]	1.00

Table 4: Constraint weights, UR probabilities, and SR likelihoods after 0-disparity EM learning.

0-disparity URs.

4.4 Step (4): UR Candidate Expansion

Unsatisfied with this learning result, the UR PHaSE learner proceeds to consider more abstract 1-disparity URs as potential additions to the UR candidate space. Importantly, 1-disparity URs are generated for all morphemes the learner encounters, not only those that fail to meet the likelihood threshold. Expanding the hypothesis spaces of all morphemes prevents the learner from cementing the 0-disparity analysis into part of the lexicon while allowing abstraction elsewhere, which can distort subsequent EM updates.

For instance, at the end of EM learning with concrete 0-disparity URs in section 4.2, the learner is unable to simultaneously model the presence of nasal harmony in alternations like [sɑ]∼[sɑ̃ʊ̃ɑ̃] 'breath'∼'breaths' and the lack of harmony in pre-N forms like [tʌvɑ̃n] 'penalty'. As a compromise, the learner lowers the weights of the constraints enforcing harmony and, in doing so, predicts the absence of harmony in pre-N forms with perfect probability, as shown in Table 4. If only the candidate spaces of morphemes with low likelihood were expanded, the learner would preserve the incorrect URs assigned to pre-N forms during the 0-disparity stage, thereby preventing it from ever arriving at the correct abstract, underlyingly oral URs for pre-N forms.

In generating 1-disparity URs, the learner first examines the violation profiles of all markedness constraints with respect to the observed surface forms at the conclusion of 0-disparity learning (Table 4). If a markedness constraint exhibits a non-maximal weight and is violated at least once by an observed SR, the feature associated with that markedness constraint is eligible to be flipped to construct new 1-disparity URs. A violated markedness constraint that fails to reach its upper bound indicates that weight re-optimization alone cannot explain the relevant patterns across SRs under current UR assumptions. The learner, therefore, treats such intermediate weights as evidence that the UR may be mis-specified along the associated feature and permits UR abstraction in order to test whether an alternative UR yields higher likelihood.

In fact, only SPRD-L[NAS], *NASG, and *NASV satisfy both criteria, rendering [NAS] as the only feature eligible to be flipped to generate new 1-disparity candidate URs. Crucially, *LOWRD both reaches the upper weight bound of 100 and is not violated by any observed SRs. Consequently, the learner determines that introducing URs with a [RD] disparity is unlikely to improve likelihood. Thus, the expansion of the UR candidate space with the addition of 1-disparity URs will be relatively modest, only introducing URs that flip the [NAS] feature on eligible segments.

Table 5 shows the 1-disparity URs added to the hypothesis spaces of the morphemes 'breath' and 'penalty'. The two morphemes differ sharply in how their UR candidate spaces expand. For 'breath', the candidate space remains unchanged because flipping the [NAS] feature on the vowel yields URs that already belong to the 0-disparity level. Furthermore, nasalized [s̃] is unattested in Punjabi, leaving the learner without a phonological category that could support representations such as /s̃ɑ̃/ or /s̃ɑ/. In contrast, the UR candidate space for 'penalty' expands from the single SR to four potential URs. By flipping the [NAS] feature on all segments for which a corresponding segmental category exists, the learner introduces a small but analytically consequential set of 1-disparity URs. Specifically, the expanded UR space includes analytically correct UR, /tʌvɑn/, while remaining tightly constrained and computationally tractable.

Morpheme	UR Candidate Space
'breath'	/sɑ/ /sɑ̃/
'penalty'	/tɑvɑ̃n/ /tɑṽɑ̃n/ /tɑ̃vɑ̃n/ /tɑvɑn/

Table 5: UR candidates for 'breath' and 'penalty', expanded to include all possible 1-disparity URs generated by flipping the [NAS] feature.

Constraint	Type	initial w	final w
SPRD-L[NAS]	mark.	0.13	48.45
*NASOBS	mark.	100.00	100.00
*NASG	mark.	0.47	48.45
*NASV	mark.	0.00	0.00
*VN	mark.	100.00	100.00
ID[NAS]	faith.	0.00	0.00
IDFin[NAS]	faith.	0.00	100.00
ID[NAS]/_V	contfaith.	0.00	100.00
ID[RD]	faith.	0.00	6.28
*LOWRD	mark.	100.00	100.00

Morpheme	UR candidate	Prior	Posterior
'breath'	/sɑ/	0.5	1.00
	/sɑ̃/	0.5	0.00
'penalty'	/tɑvɑ̃n/	0.25	0.00
	/tɑvɑn/	0.25	1.00
	/tɑṽɑ̃n/	0.25	0.00
	/tɑ̃vɑ̃n/	0.25	0.00

SR	Likelihood
[sɑ]	1.00
[sɑ̃ṽɑ̃]	0.96
[tɑvɑ̃n]	1.00

Table 6: Constraint weights, UR probabilities, and SR likelihoods after 1-disparity EM learning.

4.5 Step (2): 1-disparity UR EM Learning

Equipped with the newly expanded hypothesis spaces, the learner returns to step (2) to reinstantiate EM learning, the results of which are shown in Table 6. As is evident, the combination of learned weights and UR probability distributions enables the learner to predict the observed SRs with high degrees of likelihood. This is achieved by assigning all of the UR probability for 'breath' to oral /sɑ/ and deriving nasality on the stem by way of nasal harmony triggered by the vowel in the plural suffix. Conversely, the learner also correctly assigned all of the probability to the abstract UR /tɑvɑn/ for 'penalty', thereby enabling the pre-N vowel to surface as nasal (due to high weight *VN) while not predicting nasal harmony on these forms (due to high weight on the contextual faithfulness constraint, ID[NAS]/_V).

4.6 Step (3): Likelihood Threshold Eval. for 1-disparity URs

With the addition of 1-disparity URs, the learner converged on a combination of constraint weights and UR probabilities that yielded a likelihood greater than .95 for each SR, thus successfully acquiring a grammar and lexicon that accounts for the data.

5 Discussion & Future Research

This paper introduced the UR Progressive Hypothesis Space Expansion (UR PHaSE) learner, a novel learning architecture designed to reconcile the empirical need for abstract URs with the learnability concerns raised by permitting increased UR abstraction. Rather than evaluating an effectively unbounded UR hypothesis space in parallel, the learner structures that space by disparity count and searches it serially, beginning with fully concrete (0-disparity) URs and expanding to include more abstract candidates only when the current level fails to meet a Likelihood Threshold.

The UR PHaSE learner was applied to a Pakistani Punjabi vowel nasality pattern that requires abstract URs to account for pre-N vowels, showing that 0-disparity EM learning predictably fails to acquire the Punjabi nasality patterns because concrete URs cannot simultaneously induce harmony in contrastive-nasal contexts while suppressing it in pre-N contexts. The learner then selectively expands the UR candidate spaces using information already present in the learned grammar to generate only those higher-disparity URs that are plausibly informative, keeping hypothesis space growth tractable. More specifically, when learning fails at lower levels of abstraction, the learner uses the constraints that highlight uncertainty/inconsistency in the grammar (i.e., constraints with non-maximal weights) to collect a set of features that could be changed in the representations to resolve this uncertainty. With this targeted expansion, the learner successfully converges on the minimally abstract URs required by the Punjabi pattern, thereby demonstrating that restrictedly abstract URs are learnable when hypothesis space expansion is progressive, structured, and threshold-governed.

The UR PHaSE learner developed in this paper shows that constraints on UR abstraction need not be imposed as an external stipulation. Building on the conception of structuring UR hypothesis spaces by disparity count proposed by Tesar (2014,

2016), it avoids evaluating an unbounded set of candidates in parallel by generating and considering URs in batches, based on their disparity distance. This architecture naturally restricts abstraction because minimally abstract URs are always preferred if they are sufficient. At the same time, the learner addresses search space concerns presented by UR abstraction by expanding the hypothesis space in a targeted way. As the Punjabi example shows, this combination allows the learner to arrive at the highly abstract URs needed for pre-N vowels.

Several directions for future work follow from these results. First, the present paper demonstrates success on a single pattern that motivates UR abstraction, Pakistani Punjabi vowel nasality. The learner needs to be tested on additional case studies that have been argued to require abstract URs (e.g., O'Hara (2017) and Wang and Hayes (2025)). Second, because the UR PHaSE learner does not impose an explicit upper bound on abstraction, it predicts that if a pattern truly requires 2-disparity or 3-disparity URs, the learner will eventually incorporate those URs into the candidate space. At the same time, after the learner has climbed deep enough into the disparity ladder, the UR candidate space may still grow too large, ultimately yielding prohibitive compute time or convergence failures. If so, then we should not expect to find stable phonological systems in which very high degrees of UR→SR mapping disparities are analytically necessary. Exploring where this boundary lies could clarify how far abstraction can plausibly extend under realistic learning constraints and why, over time, learners tend to reanalyze highly abstract URs as concrete (Kiparsky, 1973; Kuo, 2024a,b).

In sum, the UR PHaSE learner offers a principled way to reconcile abstract UR analyses with learnability. By structuring the UR hypothesis space and searching it serially, the UR PHaSE learner both restricts abstraction to the extent necessary and preserves tractability, thereby removing the need for an externally imposed ceiling on UR abstraction.

References

Caleb Belth. 2026. A learning-based account of phonological tiers. *Linguistic Inquiry*, 57(2):229–265.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. In *Royal Statistical Society*.

Madeline Gilbert. 2023. Testing for underlying representations: Segments and clusters in Sevillian Spanish. *Natural Language & Linguistic Theory*, 42:493–531.

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. 2009. Restrictiveness and phonological grammar and lexicon learning. In *Proceedings of the 43rd Annual Meeting of the Chicago Linguistics Society*, volume 43, pages 125–134.

Gaja Jarosz. 2010. Implicational markedness and frequency in constraint-based computational models of phonological learning. *Journal of Child Language. Special Issue on Computational Models of Child Language Learning*, 37(3):565–606.

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.

Paul Kiparsky. 1973. *Abstractness, opacity, and global rules*, pages 57–86. Tokyo: TEC.

Jennifer Kuo. 2024a. Phonetic naturalness in the reanalysis of Samoan thematic consonant alternations. *Journal of Phonetics*, 107:1–18.

Jennifer Kuo. 2024b. Phonological reanalysis is guided by markedness: the case of malagasy weak stems. *Phonology*, 41(3):1–35.

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(2):325–345.

Jonathan Charles Paramore. 2025. Learning covert URs via disparity minimization. In *Eighth Annual Meeting of the Society for Computation in Linguistics (SCiL)*, volume 8.

Jonathan Charles Paramore and Ryan T. Bennett. 2025. Covert URs: evidence from nasalization in Pakistani Punjabi. Manuscript, UC Santa Cruz.

Alan Prince and Paul Smolensky. 1993/2004. *Optimality Theory: constraint interaction in Generative grammar*. Malden, Mass: Blackwell Publishers.

Herbert A. Simon. 1955. A behavioral model of rational choice. *The Quarterly Journal of Economics*, 69(1):99–118.

Herbert A. Simon. 1956. Rational choice and the structure of the environment. *Psychological Review*, 63(2):129–138.

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*, pages 1–39.

A Appendix

(1) Surface forms provided to the learner

i. **CV**: [sɑ], [do], [to], [pe], [ge], [si]

ii. **CṼ-G̃Ṽ**: [sɑ̃ṽɑ̃], [dõṽɑ̃], [tõṽɑ̃], [pẽṽɑ̃], [gẽṽɑ̃], [sĩṽɑ̃]

iii. **CṼ**: [gɑ̃], [tʃʰɑ̃], [kõ], [ʒɑ̃], [kɑ̃], [põ]

iv. **CṼ-G̃Ṽ**: [gɑ̃ṽɑ̃], [tʃʰɑ̃ṽɑ̃], [kõṽɑ̃], [ʒɑ̃ṽɑ̃], [kɑ̃ṽɑ̃], [põṽɑ̃]

v. **CVGṼN**: [tɑvɑ̃n], [prəvɑ̃ɳ], [tejɑ̃ɳ], [ʒəvɑ̃ɳ], [ɑvɑ̃m], [sijɑ̃ɳ]

(2) Constraint Definitions

i. SPRD-L[NAS] (cf. Walker, 2003, p.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, p.51)
Assign a violation for every obstruent that dominates a [+NAS] feature.

iii. *NASG (Walker, 2003, p.51)
Assign a violation for every glide that dominates a [+NAS] feature.

iv. *NASV (Walker, 2003, p.51)
Assign a violation for every vowel that dominates a [+NAS] feature.

v. *VN
Assign a violation for every vowel that dominates a [-NAS] feature when directly preceding a nasal consonant.

vi. 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*.

vii. 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.

viii. 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*.

ix. 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*.

x. *LOWRD
Assign a violation for every vowel that dominates a [RD] feature and a [LOW] feature simultaneously.

What Exactly do Children Receive in Language Acquisition? A Case Study on CHILDES with Automated Detection of Filler-Gap Dependencies

Zhenghao Herbert Zhou[1] **William Dai**[2] **Maya Viswanathan**[2]
Simon Charlow[1,3] **R. Thomas McCoy**[1,3] **Robert Frank**[1,3]
[1]Department of Linguistics, Yale University
[2]Department of Computer Science, Yale University
[3]Wu Tsai Institute, Yale University
{herbert.zhou, william.dai, maya.viswanathan,
simon.charlow, tom.mccoy, robert.frank}@yale.edu

1 Motivation

Language learners generalize beyond the surface strings they hear, but theories posit various factors on what supports this generalization. The *poverty-of-stimulus* view posits innate biases that outstrip available evidences, while statistical learning accounts emphasize how learners exploit distributional regularities in caregiver input. Filler-gap dependencies (FGDs) are a useful testbed because they span multiple constructions (e.g., wh-questions and relative clauses) and exhibit extraction-site asymmetries (e.g., subject vs. object gaps) that affect learning and processing. Yet the relevant input is hard to quantify at scale with the granularity needed to adjudicate these accounts.

In this study: (i) we present an automated, fine-grained detection tool for three FGD constructions subtyped by extraction sites; (ii) we apply it to the entire CHILDES database to characterize FGD distributions children receive and produce; and (iii) we outline how our tool can be applied to both address open questions in acquisition and to study linguistic generalization in modern language models.

2 Target Constructions and Our Detector

We present an automated detector that identifies three core spoken-English FGD families in CHILDES —matrix questions (MQs), embedded questions (EQs), and relative clauses (RCs) — and subtypes each instance by extraction site (subject vs. object vs. adjunct; plus polar and reduced variants where applicable). Our detector leverages the strengths of both constituency parsing (to extract clausal boundaries and complement types) and dependency parsing (to access head-dependent configurations informative about extraction sites). For each construction, we: (i) detect a local constituency signature (e.g., `NP` → `NP SBAR` for RCs); (ii) retrieve the wh-phrase's category (e.g., `WHNP`); (iii) infer the likely gap site using structural cues inside the clause; and (iv) validate the hypothesized label with dependency relations (e.g., `relcl`, `nsubj`, `dobj`). Combining both parsing strategies improves robustness by handling cases where either parse alone is systematically insufficient, and by reducing false detections caused by noisy and error-prone speech transcriptions.

3 Evaluation

We evaluate our detector by comparing against the manually annotated treebank from Pearl and Sprouse (2013), which contains 56,461 child-directed utterances from two CHILDES corpora, annotated with trace information.

Category	Precision (total)	Recall (total)	F1
SMQ	0.827 (712)	0.792 (716)	**0.809**
OMQ	0.908 (4950)	0.977 (4464)	**0.941**
AMQ	0.921 (1839)	0.957 (1746)	**0.939**
SEQ	0.925 (146)	0.894 (236)	**0.909**
OEQ	0.958 (642)	0.895 (1325)	**0.925**
AEQ	0.808 (339)	0.905 (924)	**0.854**
PEQ	1.000 (243)	0.905 (611)	**0.950**
SRC	0.924 (157)	0.883 (247)	**0.903**
ORC	0.901 (121)	0.810 (473)	**0.853**
ARC	0.842 (38)	0.713 (94)	0.772

Table 1: Precision (against parser labels), recall (against annotation labels), and F1 scores (bolded if > 0.8) by construction and extraction site.

Precision and recall scores are shown in Table 1. Across most categories, the detector achieves strong agreement with human judgments and yields balanced precision and recall when compared to trace-derived gold labels. Despite being imperfect, our automated detector is viable for large-scale, fine-grained detection of our target structures.

4 A Case Study on CHILDES

We apply our detector to 57 English-NA CHILDES corpora accessed via `childes-db`, yielding 2.84 millions of utterances after filtering missing ages. We summarize: (i) developmental distributions (adult input vs. child production) for all utterances;

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 57–58
July 3-4, 2026 ©2026 Association for Computational Linguistics

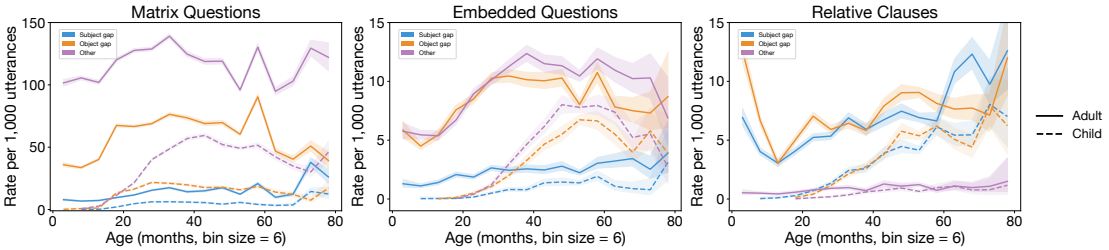


Figure 1: Adult (solid) and child (dashed) speech distributions across time by constructions and extraction sites. Uncertainty is shown as 95% Wilson intervals (wider in sparser bins).

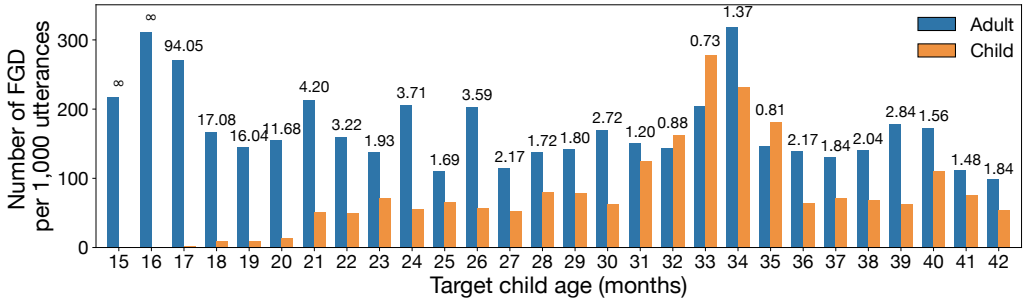


Figure 2: Per-1,000 utterance rates of all filler-gap dependency sentences received and produced by Laura.

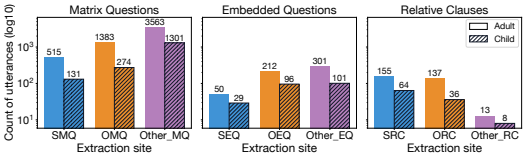


Figure 3: Number of utterances with targeted constructions received and produced by Laura in log scale.

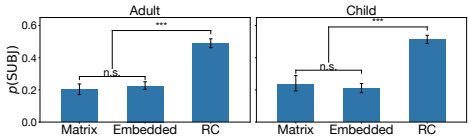


Figure 4: Cross-construction comparisons of subject share across age.

(ii) distributions of total exposures of one individual child Laura (75,740 utterances available from a longitudinal study); and (iii) extraction-site asymmetries across constructions. Three robust trends emerge: (1) matrix questions are substantially more frequent than embedded questions and relatives (Figure 1 and 3); (2) within question families, object gaps outnumber subject gaps, whereas relatives are closer to balanced (Figure 4 compares the subject share $p(\text{SUBJ}) = \frac{\#\text{SUBJ}}{\#\text{SUBJ}+\#\text{OBJ}}$ across the three constructions); and (3) child production broadly tracks adult input for question families, while relative clauses increase more gradually (Figure 1, 2, and 3). These large-scale counts offer a descriptive baseline for "what children hear" and "what children say" at a construction-by-gap-site resolution.

5 Prospective Applications

For open questions in acquisition, our detector enables frequency analyses at the level where frequency is predicted to matter (construction vs. extraction site vs. lexical combinations), supporting sharper tests of complexity-based vs. distributional accounts and role-sensitive comparisons across dependency types (Ambridge et al., 2015). For computational learners, our detector enables targeted input control for testing generalization in language models, including filtered-corpus training (Patil et al., 2024) to probe transfer across related FGDs. It also supports input-attribution analyses (Koh and Liang, 2017) that connect a model's behavior to the subsets of training data most responsible for it.

References

Ben Ambridge, Evan Kidd, Caroline F Rowland, and Anna L Theakston. 2015. The ubiquity of frequency effects in first language acquisition. *Journal of Child Language*, 42(2):239–273.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR.

Abhinav Patil, Jaap Jumelet, Yu Ying Chiu, Andy Lapastora, Peter Shen, Lexie Wang, Clevis Willrich, and Shane Steinert-Threlkeld. 2024. Filtered corpus training (FiCT) shows that language models can generalize from indirect evidence. *Transactions of the Association for Computational Linguistics*, 12:1597–1615.

Lisa Pearl and Jon Sprouse. 2013. Computational models of acquisition for islands. *Experimental Syntax and Islands Effects*, pages 109–131.

Determinants of Hesitations and Repetitions in Hindi Spontaneous Speech

Eashani Sharma* **and Ishita Arun*** **and Samar Husain**
Indian Institute of Technology Delhi
eashani.sharma@hss.iitd.ac.in
huz248352@iitd.ac.in
samar@hss.iitd.ac.in

Abstract

This study investigates the factors that predict disfluencies in Hindi spontaneous speech. In particular, we probe the influence of lexical, syntactic, phonological, and prosodic factors on two kinds of disfluencies, namely, hesitations and repetitions. These disfluencies are probed through both the nature of linguistic factors as well as through the source (preceding vs. following word) of these factors. Our results show that hesitations and repetitions pattern differently during spontaneous speech. Hesitations increase due to lexical, syntactic, as well as articulatory features from both preceding and following words. On the other hand, repetitions arise mainly due to lexical and articulatory factors of the upcoming word. Further, while previous research (e.g., Bell et al., 2009; Dammalapati et al., 2021) on English highlights the importance of upcoming difficulty on disfluencies, our results suggest that previously encountered difficulties can also lead to an increase in disfluencies. This suggests that language typology (SVO vs SOV) can play a critical role in determining the planning process and thereby affecting the distribution of disfluencies in a language. Together, these findings highlight the need for increased cross-linguistic research to understand the nature of incrementality and monitoring of the production system cross-linguistically.

1 Introduction

The conversion of thought to speech is complex and involves cognitive, linguistic, and motor faculties. It is commonly assumed that one transforms the conceptual message into a linguistic formation, assigning the lexical counterparts of the concepts and incorporating these lexical items into a syntactic structure. The output of this formulation then undergoes phonological encoding, followed by motor planning for speech output (Garrett, 1975; Bock and Levelt, 1994). While much of the speech is error-free, there is also a substantial amount of disfluencies, pauses, and repairs (Hockett, 1967; Nooteboom, 1980; Fromkin, 1971, 1973; Brennan and Williams, 1995; Lickley, 2017). Any interference or difficulty in the production process, such as delays in planning (Clark and Fox Tree, 2002), lexical retrieval, and phonological difficulties (Fraundorf and Watson, 2013; Levelt, 1983; Goldman-Eisler, 1961), a change in discourse status (Arnold et al., 2003), processing load (Bortfeld et al., 2001), or repairs (Levelt, 1983; Postma and Kolk, 1993; Blackmer and Mitton, 1991), can lead to disfluencies. In fact, it has been suggested that around 6-10% words in every 100 words are disfluent (Fox Tree, 1995).

Given that disfluencies tend to have no relation to the message's central proposition (Bortfeld et al., 2001; Fraundorf and Watson, 2013; Cossavella and Cevasco, 2021; Shriberg, 1994), they have traditionally been excluded from dedicated investigations. Under this view, they reflect production difficulties rather than linguistic meaning (Levelt, 1983; Goldman-Eisler, 1961). However, recent evidence suggests that fillers like *'Uh'* and *'Um'* hold lexical status (Clark and Fox Tree, 2002; Arnold et al., 2007). The multi-stage nature of speech production and the distributed occurrence of disfluencies in natural speech make it difficult to isolate their underlying causes: do they arise from lexical, syntactic, or any other processing bottlenecks?

Some attempts have been made in English to investigate this using corpus-based methods. The existing literature on the Switchboard corpus has revealed the following distributional patterns (particularly for filled pauses): disfluencies are likely to occur before less predictable words (Arnold et al., 2007; Sen, 2020). Words preceding disfluencies display low lexical surprisal and syntactic complexity, while words following disfluencies exhibit high lexical surprisal and syntactic complexity

*Equal contribution.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 59–71
July 3-4, 2026 ©2026 Association for Computational Linguistics

(Dammalapati et al., 2021). On the prosodic front, words neighboring disfluencies are longer or show elongated duration (Dammalapati et al., 2021), often accompanied by a prosodic flattening (Zayats and Ostendorf, 2019; Székely et al., 2017), indicating increased processing load (Ferreira, 1993). While these lexical, syntactic, and prosodic factors target different levels of sentence processing, they fall short in two respects: they do not account for typologically different languages where processing strategies may differ, and they largely overlook the role of phonological planning.

We extend previous work by systematically addressing the causes of disfluencies at different levels of sentence processing in Hindi, a Subject-Object-Verb (SOV) language. Disfluencies manifest in various forms: hesitations, repetitions, self-corrections, false starts, and reparanda (Shriberg, 1994; Fox Tree, 1995, 2001). We focus specifically on two types: hesitations and repetitions. Hesitations are interruptions in speech marked by either empty pauses (silent breaks) or filled pauses (such as *'Uh'* and *'Um'*) (Fox Tree, 2001; Clark and Fox Tree, 2002). Because it is difficult to reliably differentiate between fluent and disfluent empty pauses (Fox Tree, 1995), we focus our analysis on filled pauses and refer to them as *hesitations* hereafter. Repetitions involve the reiteration of a word or phrase (Fox Tree, 2001; Clark and Fox Tree, 2002; Levelt, 1983).

To identify the locus of difficulty during sentence production, we examine both preceding and following words across four processing levels using the following features:

1. **Lexical**: frequency, semantic similarity, and bigram probability
2. **Structural**: integration cost and surprisal
3. **Phonological**: word length[1] and phonological complexity
4. **Prosodic**: duration, pitch, and intensity

We address predictability at the lexical and syntactic levels using surprisal (Hale, 2001; Levy, 2008), semantic similarity (Miller and Charles, 1991; Roland et al., 2012), and bigram probabilities (Jurafsky and Martin, 2013); structural complexity with integration cost (Gibson et al., 2000); and phonological complexity with word length and phonotactic complexity (Vitevitch and Luce, 1999; Vitevitch et al., 2004; Celata, 2020). The associated cognitive effort is informed by prosodic analysis on a subset of the data. This approach offers a comprehensive understanding of disfluencies.

The paper is organized as follows: Section 2 introduces our dataset and discusses analyses at different linguistic levels, along with our predictions. Section 3 presents the results of the mixed-effects models for hesitations and repetitions, including prosodic predictors on the hesitation data subset. Section 4 discusses our findings and their implications. Section 5 concludes the paper.

2 Methods

2.1 Dataset

We use the IIT Delhi Hindi Dialog Corpus (Pareek et al., 2025), a collection of 60 unscripted spoken telephonic conversations from the Hindi segment of the CALLFRIEND Project (Canavan and Zipperlen, 1996). This gold standard corpus contains 39,244 sentences that have been transcribed and tagged for universal Parts of Speech (POS) categories and dependency relations.

Additionally, different types of disfluencies are encoded at the token level in the CoNLL-U format using specific tags stored in the miscellaneous (MISC) column. We focus on two such tags, specifically, filled pauses that are marked with a `Hesitation` tag, and repetitions that are marked with a `Disfluency` tag on the corresponding token (see Figure 1). In our analysis, we refer to these as hesitations and repetitions, respectively. We extracted all sentences containing at least one of these tags, yielding 2,289 utterances with hesitations (5.8%) and 1,379 utterances with repetitions (3.6%).

2.2 Analysis

The analysis proceeded in several stages. First, sentences containing hesitations and repetitions were extracted using the CONLL-U parser in Python (Buchholz and Marsi, 2006). For each target sentence, we identified the words immediately preceding and following hesitations or repetitions and extracted their linguistic features. We also extracted control sentences without speech errors, matched for sentence length, to serve as a baseline for fluent speech. Each target sentence was paired with one control sentence such that the control token occurred at the same index position as the hesitation

[1]Word length, measured as syllable count, was categorized as a phonological predictor, as a greater number of syllables imposes greater phonological planning demands.

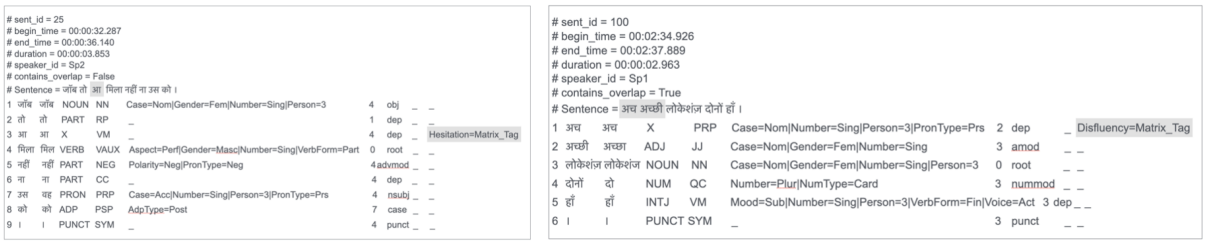


Figure 1: Example sentences from the IIT Delhi Hindi Dialogue Corpus in CoNLL-U format with transcription-level annotations and non-linguistic metadata. Hesitation (left): the hesitation token and its corresponding annotation in the miscellaneous column are highlighted in gray. Repetition (right): the disfluent token and its corresponding annotation are highlighted in gray.

or repetition token. Finally, we fitted mixed-effects logistic regression models using the lme4 package in R (Bates et al., 2015).

2.2.1 Linguistic features

Since disfluencies are widely understood to reflect difficulties at different stages of speech production, including lexical retrieval (Levelt, 1983, 1989), syntactic planning (Dammalapati et al., 2021; Bock and Levelt, 1994; Scontras et al., 2015), and articulation (Bell et al., 2009; Székely et al., 2017), difficulty at any stage can surface locally in the speech signal (Clark and Fox Tree, 2002; Levelt, 1983). Motivated by this, we investigated the following linguistic features in words immediately preceding or following hesitations and repetitions, as well as matched control positions, to capture production difficulty across multiple processing levels.

- **Word Frequency:** An estimate of how frequently a word occurs in the corpus. To normalize this value, the raw frequency count was Zipf-transformed.
- **Surprisal:** A measure of the unexpectedness of a word in a context, computed as the negative logarithm of the probability of the word given the preceding context (Hale, 2001; Levy, 2008).
- **Semantic similarity:** An estimate of how semantically related a word is to its preceding context, computed as the cosine similarity between the word's embedding and a vector representation of the preceding context (Miller and Charles, 1991; Roland et al., 2012).
- **Integration cost:** A measure of syntactic complexity, computed as the sum of linear distances between the target word and each of its linearly preceding head/dependents, with an additional baseline cost of 1 (Gibson et al., 2000).
- **Bigram probability:** The probability of a word given the preceding word. When a bigram was unavailable, a fallback to unigram probabilities was applied (Jurafsky and Martin, 2013).
- **Word length:** The number of syllables in a word.
- **Phonological complexity:** A measure of phonotactic complexity, computed as the probability of a consonant–vowel (CV) sequence given the preceding CV patterns found in the words of the corpus.

Details of the computational implementation of each feature are provided in Appendix A.

After feature extraction, we performed several pre-processing steps in R (R Core Team, 2025). We removed duplicate sentences and cases with missing values for any predictor. We also excluded sentences with hesitations or repetitions in sentence-initial or sentence-final positions, as these do not permit analysis of both preceding and following lexical contexts.

For hesitations, we obtained 2,095 unique sentences from the corpus. After excluding 673 sentences with sentence-initial hesitations (32%) and 97 with sentence-final hesitations (4.6%), and removing cases with missing values, we retained 897 sentences containing hesitations. These were paired with 897 matched control sentences, yielding a final dataset of 1,794 sentences.

For repetitions, we obtained 1,379 unique sentences. After excluding 329 sentences with repetitions at the sentence onset (23.86%), 23 at the sentence end (1.67%), and removing rows with missing values, we retained 780 sentences. These were paired with 780 matched control sentences, yielding a final dataset of 1,560 sentences.

We fitted separate generalized linear mixed-effects (*glmer*) models for hesitations and rep-

etitions, with a binary outcome variable (presence/absence of disfluency) as the dependent variable. Both models included linguistic features with only weak to moderate correlations[2] of the preceding and following words as fixed-effect predictors, and conversation ID[3] as a random intercept.

The hesitation model included an additional interaction term between preceding word frequency and length. The repetition model excluded word length[4] as a predictor and instead included an interaction between following word frequency and semantic similarity. Model selection followed standard recommendations for generalized linear mixed-effects models, including convergence checks and the comparison of nested random-effects structures whenever necessary (Winter, 2019; Barr et al., 2013).

$$\begin{aligned}\texttt{Hesitation} \sim\ & \texttt{WordFreq}_{\text{prev}} + \texttt{WordFreq}_{\text{next}} + \texttt{IC}_{\text{prev}} \\ & + \texttt{IC}_{\text{next}} + \texttt{SemSim}_{\text{prev}} + \texttt{SemSim}_{\text{next}} \\ & + \texttt{Surprisal}_{\text{prev}} + \texttt{Surprisal}_{\text{next}} \\ & + \texttt{BiProb}_{\text{prev}} + \texttt{BiProb}_{\text{next}} \\ & + \texttt{WordLen}_{\text{prev}} + \texttt{WordLen}_{\text{next}} \\ & + \texttt{PhonComp}_{\text{prev}} + \texttt{PhonComp}_{\text{next}} \\ & + \texttt{WordFreq}_{\text{prev}} * \texttt{WordLen}_{\text{prev}} \\ & + (1 \mid \texttt{ConversationID})\end{aligned}$$

$$\begin{aligned}\texttt{Repetition} \sim\ & \texttt{WordFreq}_{\text{prev}} + \texttt{WordFreq}_{\text{next}} + \texttt{IC}_{\text{prev}} \\ & + \texttt{IC}_{\text{next}} + \texttt{SemSim}_{\text{prev}} + \texttt{SemSim}_{\text{next}} \\ & + \texttt{Surprisal}_{\text{prev}} + \texttt{Surprisal}_{\text{next}} \\ & + \texttt{BiProb}_{\text{prev}} + \texttt{BiProb}_{\text{next}} \\ & + \texttt{PhonComp}_{\text{prev}} + \texttt{PhonComp}_{\text{next}} \\ & + \texttt{WordFreq}_{\text{next}} * \texttt{SemSim}_{\text{next}} \\ & + (1 \mid \texttt{ConversationID})\end{aligned}$$

2.2.2 Prosodic features

To examine the prosodic context surrounding hesitations, we sampled 130 sentences with 6 to 16 tokens containing hesitations and 130 control sentences without any speech errors, matched for sentence length. For sentences without hesitation, we aligned the token positions with those of the hesitations to ensure the parallel extraction of the preceding and following words for comparative analysis. These sentences were annotated in PRAAT version 6.4.26 (Boersma, 2012) at the sentence and word levels and exported to TextGrid files. A custom PRAAT script was used to extract pitch, duration, and intensity measurements for the target sentences, as well as for the preceding and following words.

Statistical analysis was conducted at the sentence and word levels in R. First, we examined how sentence-level prosodic properties could predict the presence of hesitations by fitting a *glmer* model with a binary outcome variable indicating the presence or absence of hesitations. Fixed-effect predictors included sentence duration, average pitch, and average intensity, all standardized prior to modeling. To capture joint effects of temporal and prosodic variation, the model additionally included interactions between duration and average pitch, and between average pitch and sentence length. Speaker [5] was included as a random intercept.

$$\begin{aligned}\texttt{Hesitation} \sim\ & \texttt{Duration} + \texttt{Intensity} + \texttt{Pitch} \\ & + \texttt{Duration} * \texttt{Pitch} \\ & + \texttt{Pitch} * \texttt{SentenceLength} \\ & + (1 \mid \texttt{Speaker})\end{aligned}$$

For the word-level analysis, we investigated how the prosodic features in the immediate context varied as a function of hesitations. Sum contrasts were applied to two categorical predictors, hesitation (present vs. absent) and token position (preceding vs. following), to estimate their main effects and interaction. Separate linear mixed-effects models were fitted for duration, pitch, and intensity, with hesitation and token position specified as fixed-effect predictors. Speaker, gender[6], and sentence length were included as random intercepts to account for participant, demographic, and utterance-level variability. Model selection followed standard recommendations for linear mixed-effects models, including convergence diagnostics and comparisons of nested random-effects structures (Winter, 2019; Barr et al., 2013).

2.3 Predictions

Under the assumption that previous findings would hold cross-linguistically, we hypothesized that hesitations and repetitions would be more likely

[2] All continuous predictors were z-scored prior to model fitting. Phonological complexity was log-transformed before standardization to reduce skew.

[3] Conversation ID uniquely identifies each telephone conversation in the corpus (corresponding to the recording filename). It serves as a discourse-level random effect in our mixed-effects models to account for within-conversation variability.

[4] Word length was excluded as a predictor due to model non-convergence.

[5] Speaker denotes an individual participant in a recorded telephone conversation and serves as a random effect to capture participant-level variability.

[6] Speaker and gender information is available as metadata in the IIT Delhi Hindi Dialog corpus (Pareek et al., 2025).

when words preceding them have low lexical surprisal and integration costs, while words following them have higher surprisal and integration costs (Dammalapati et al., 2021; Arnold et al., 2003). Since speech errors are known to arise from an increase in planning load (Clark and Fox Tree, 2002; Levelt, 1983; Fraundorf and Watson, 2013; Arnold et al., 2003), we also predicted that hesitations and repetitions would be more likely to occur when words following them have lower word frequency and bigram probability, along with higher semantic similarity[7] and phonological complexity.

For prosodic features, we hypothesized that words preceding hesitations would have a lower pitch, while words following hesitations would have a higher pitch to mark restarts (Zayats and Ostendorf, 2019; Székely et al., 2017). Similarly, for duration, we hypothesized that words preceding hesitations would be longer due to increased planning load and retrieval delay, and that words following hesitations would be shorter as fluency is regained (Dammalapati et al., 2021). The average intensity surrounding hesitations was anticipated to differ along the same lines, typically decreasing before hesitations and rising after them.

3 Results

3.1 Linguistic predictors of hesitations

Hesitations were predicted by frequency ($p < .05$), integration cost ($p < .05$), surprisal ($p < .01$), and phonological complexity ($p < .01$) of the preceding word, along with integration cost ($p < .01$), semantic similarity ($p < .01$), word length ($p < .001$), and phonological complexity ($p < .01$) of the following word. Hesitations increased with higher surprisal, higher integration cost, and greater phonological complexity of the preceding word. In addition, hesitations increased with higher integration cost, increased semantic similarity, increased word length, and phonological complexity of the following word. A higher frequency of the preceding word reduced the likelihood of hesitations. A significant interaction between the frequency and length of the preceding word ($p < .05$) revealed that preceding word frequency increased the likelihood of hesitations more strongly after shorter words than after longer words ($\beta = 3.47$, $p = .012$); see Table 1 for all model coefficients.

[7]A higher semantic similarity indicates competition between all semantically possible activated words creating interference and subsequent selection difficulty (Levelt et al., 1999; Roelofs, 1992)

3.2 Linguistic predictors of repetitions

The frequency ($p < .05$), integration cost ($p < .05$), bigram probability ($p < .05$), and phonological complexity ($p < .001$) of the following word, as well as the phonological complexity of the preceding word ($p < .001$), significantly predicted the presence of repetitions. Repetitions were more likely to occur when both the preceding and following words were phonologically complex and when the following word had a higher bigram probability. Conversely, higher frequency and integration cost of the following word reduced the likelihood of repetitions. A significant interaction effect between the frequency and semantic similarity of the following words ($p < .01$) revealed that frequency reduced repetitions more strongly in high-similarity contexts than in low-similarity contexts ($\beta = 18.8$, $p = .001$).

3.3 Prosodic predictors of hesitations

At the sentence level, duration significantly predicted hesitation ($p < .05$), indicating that for sentences of the same length, longer durations are associated with a higher likelihood of hesitation. Average pitch ($p = .67$) and average intensity ($p = .64$) of the sentence did not significantly predict hesitations. However, descriptive analysis of time-normalized pitch contours at 10% intervals revealed consistently lower pitch at points where hesitations occurred (see Figure 6 in the Appendix).

We then compared the prosodic properties of words before and after a hesitation, as well as with words corresponding to those positions in fluent sentences. Hesitations significantly increased the duration of the surrounding words ($p < .001$) (see Figure 2). The main effect of word position ($p = .62$) and the hesitation $\times$ position interaction ($p = .10$) were not significant. Hesitation tokens themselves were significantly longer than their matched fluent counterparts (hesitations: M = 301 ms, SE = 10.9; fluent: M = 225 ms, SE = 9.0; $\beta = 0.31$, $p < .001$) (see Figure 7 in the Appendix).

Pitch was analyzed using an *lmer* model with random intercepts for speaker, gender, and sentence length. No significant effects were found for hesitations ($p = .14$) and position ($p = .28$). A significant hesitation $\times$ position interaction was found for pitch ($p = .02$), driven by a decrease in pitch in the word following hesitations, though the pairwise difference was not significant ($p = .082$) (see Fig-

Predictor	Hesitation				Repetition			
	Coef	SE	z	p	Coef	SE	z	p
Preceding word frequency	-2.97	1.16	-2.56	0.0106*	-0.91	0.53	-1.73	0.084
Following word frequency	-0.52	1.00	-0.53	0.600	-1.65	0.69	-2.40	0.0165*
Preceding integration cost	2.95	1.21	2.44	0.0149*	-0.00	0.32	-0.01	0.994
Following integration cost	1.05	0.40	2.65	0.0080**	-0.49	0.25	-1.99	0.0468*
Preceding surprisal	1.94	0.72	2.70	0.0069**	-0.06	0.41	-0.15	0.884
Following surprisal	-0.88	0.70	-1.26	0.208	-0.63	0.48	-1.32	0.188
Preceding semantic similarity	1.65	1.10	1.49	0.135	-0.33	0.44	-0.74	0.462
Following semantic similarity	3.73	1.27	2.93	0.0034**	0.08	0.49	0.16	0.872
Preceding bigram probability	-0.76	0.45	-1.69	0.090	-0.48	0.28	-1.69	0.090
Following bigram probability	-1.29	0.70	-1.84	0.066	1.10	0.43	2.54	0.0112*
Preceding word length	1.92	1.09	1.76	0.079	-	-	-	-
Following word length	4.09	1.24	3.31	0.0010***	-	-	-	-
Preceding phonological complexity	11.87	3.67	3.23	0.0012**	5.65	1.13	4.99	< 0.001***
Following phonological complexity	8.30	2.58	3.22	0.0013**	6.73	1.19	5.68	< 0.001***
Preceding word frequency × word length	-2.53	1.01	-2.52	0.0118*	-	-	-	-
Following word frequency × semantic similarity	-	-	-	-	-2.33	0.73	-3.21	0.0013**

*p<0.05, **p<0.01, **p<0.001

Table 1: Fixed effects from *glmer* models predicting hesitations and repetitions. Shaded rows indicate statistically significant effects.

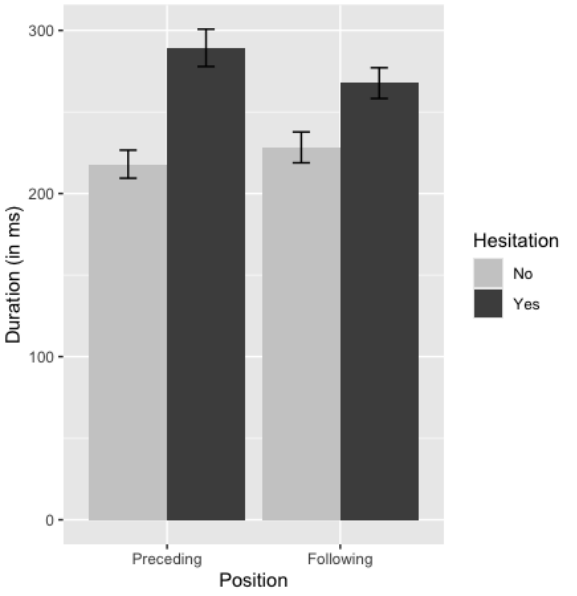


Figure 2: Mean duration of words preceding and following hesitations compared to words in no hesitation contexts.

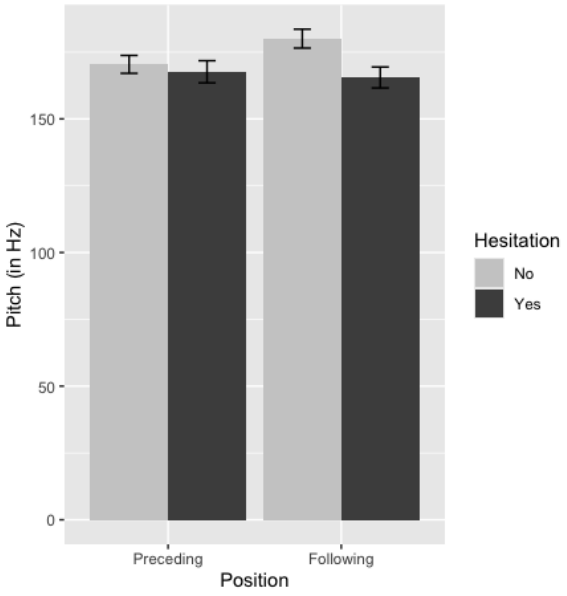


Figure 3: Mean pitch of words preceding and following hesitations compared to words in no hesitation contexts.

ure 3). However, the average pitch of hesitations was significantly lower than that of their matched fluent counterparts (hesitations: M = 162 Hz, SE = 4.28; fluent: M = 175 Hz, SE = 3.66; $\beta = -0.12$, $p = .003$) (see Figure 8 in the Appendix).

The average intensity of words was compared using an *lmer* model, which included random intercepts for speaker, gender, and sentence length. While the presence of hesitations ($p = .10$), word position ($p = .35$), or their interaction ($p = .20$) failed to reach significance (see Figure 4), the intensity of the hesitations themselves was significantly lower than that of their matched fluent counterparts (hesitations: M = 63.7 db, SE = 0.57; fluent: M = 64.9 db, SE = 0.41; $\beta = 0.04$, $p < .01$) (see Figure 9 in the Appendix).

4 Discussion

We investigated the determinants of hesitations and repetitions in spontaneous Hindi speech using the IIT Delhi Hindi Dialog Corpus (Pareek et al., 2025). We demonstrate that both hesitations and repetitions are sensitive to lexical, syntactic, and phonological factors (see Table 1). Our findings are consistent with the previous literature on disfluencies

Prosodic Feature	Fixed Effect	Coef	*SE*	*t*	*p*
Duration	Hesitation	−52.39	12.40	−4.23	< .001**
	Position	4.77	9.60	0.50	.62
	Hesitation × Position	−31.56	19.19	−1.65	.10
Pitch	Hesitation	6.94	4.68	1.48	0.14
	Position	−2.99	2.75	−1.09	0.28
	Hesitation × Position	−12.50	5.50	−2.27	0.02*
Intensity	Hesitation	1.18	0.72	1.64	0.10
	Position	−0.39	0.41	−0.93	0.35
	Hesitation × Position	−1.07	0.83	−1.29	0.20

*p<0.05, **p<0.01, **p<0.001

Table 2: Fixed effects from *lmer* models examining prosodic predictors of hesitation. Shaded rows indicate statistically significant effects.

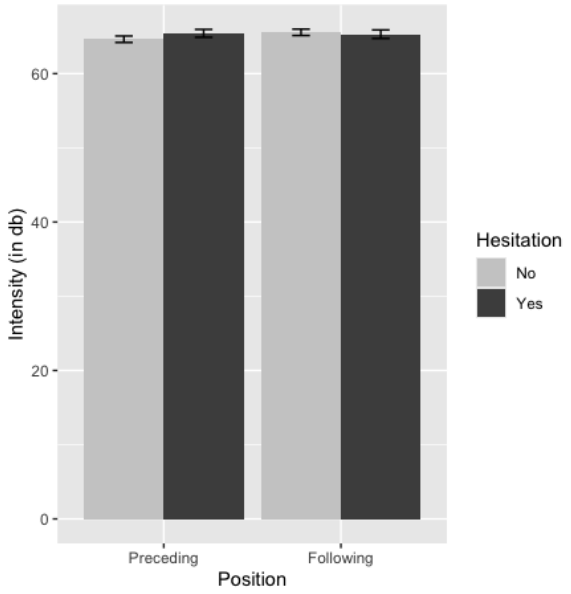


Figure 4: Mean intensity of words preceding and following hesitations compared to words in no hesitation contexts.

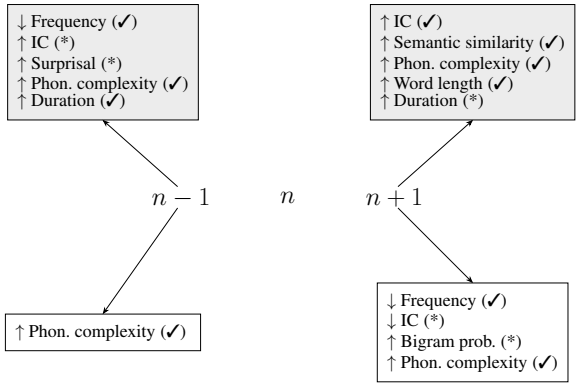


Figure 5: Summary of linguistic and prosodic predictors exhibiting statistically significant effects across position (Preceding $(n-1)$ vs. Following $(n+1)$) and speech error type (Hesitation vs. Repetition). A speech error is denoted by n. Grey panels correspond to hesitation effects, whereas white panels correspond to repetition effects. A check mark (✓) indicates a predicted effect; a star $(*)$ indicates an unexpected result.

for duration and integration cost of the following word (Dammalapati et al., 2021; Shriberg, 1994), as well as prosodic flattening (Zayats and Ostendorf, 2019; Székely et al., 2017). At the same time, our results show that the properties of preceding and following words differentially predict both disfluency types. These findings supplement the existing literature with evidence from Hindi, a typologically distinct head-final language, and reveal several salient patterns across disfluency types that we detail below.

Our first salient finding is that hesitations (as opposed to repetitions) increase when processing load at the previous word is high (see Figure 5). In particular, hesitations increase when the integration cost, surprisal, and phonological complexity of the preceding word increase, and hesitations decrease when the preceding word frequency is high. At the prosodic level, longer durations of surrounding words and notably lower pitch at hesitation sites further support the processing load hypothesis. This set of results is quite interesting because much previous work has found that the role of upcoming difficulty is more critical for hesitations (e.g., Dammalapati et al., 2021). This effect of the preceding element suggests that the processing load in the preceding context could limit the resources available for prior planning, which could reduce the scope of planning. This reduced planning scope manifests as hesitation. Given that such effects of the previous word have not been found in English (e.g., Dammalapati et al., 2021), we speculate that this could be an example of typological distinction during sentence planning. Recent research suggests that intervening dependency heads determine the syntactic complexity in a region (e.g., Yadav et al., 2022). As Hindi is a head-final language and En-

glish is head-initial, planning the resource-intensive heads that come later could affect resource allocation during production, effectively creating downstream difficulty (cf. Scontras et al., 2015; Gibson et al., 2000). In other words, an increase in difficulty encountered at a word would lead to increased resource utilization to process it, which could reduce the available resources needed to plan the upcoming linguistic material (Fox Tree, 1995).

One possible way to test this resource limitation proposal would be to employ forward surprisal (Ranjan et al., 2020), which captures the conditional probability of a word given the *following* context.[8] The idea is that if the processing difficulty affects planning, then the forward surprisal at a word would get attenuated when the preceding word difficulty is high. Results show that the critical interaction between the frequency of the preceding word and the following forward surprisal was not significant. See Table 3 in the Appendix section for more details. This preliminary analysis did not find evidence for reduced planning hypothesis as operationalized by forward surprisal. Future work should investigate this hypothesis more thoroughly by employing other metrics.

As opposed to the influence of the preceding word on hesitations, the following word properties showed a stronger alignment with previous literature (Dammalapati et al., 2021; Arnold et al., 2003), such that integration cost, semantic similarity, word length, and phonological complexity collectively predicted hesitations in the expected direction (see Figure 5).

In contrast to hesitations, for repetitions, the majority of preceding word predictors showed no effect. Critically, the fundamental distinction between hesitations and repetitions emerges in their position-sensitivity. Hesitations are predicted by properties of both preceding and following words, whereas repetitions are predicted primarily by properties of the following word. This asymmetry suggests that hesitations and repetitions arise from different stages of speech production (Fraundorf and Watson, 2013; Lickley, 2017).

Interestingly, phonological complexity was an exception to this asymmetry, predicting both hesitations and repetitions. We speculate that this is because phonological complexity imposes processing costs in two possible ways, depending on the word's position: a complex preceding word sustains demands on the perceptual loop beyond its articulation (Levelt, 1983, 1989), depleting the resources available for planning upcoming material. A complex following word disrupts the incremental attachment of segments to a metrical frame and syllabic gesture computation (Levelt, 1989). Failure at the initial slot results in hesitation, whereas a mid-word disruption forces the speaker to repeat already-articulated material, resulting in a repetition. This effect demonstrates that phonological encoding is an important constraint on speech fluency (Levelt, 1983; Levelt and Cutler, 1983). A bottleneck at the phonological level can notably lead to speech disruptions (Levelt, 1989) manifesting as hesitations and repetitions. This effect is further supported by our prosodic analysis, where we show that hesitations are surrounded by words with longer durations, which may provide additional time for planning upcoming words (Dammalapati et al., 2021; Bell et al., 2009). In contrast, the lower pitch and intensity of hesitation tokens themselves indicate reduced attention to prosodic modulation during articulation when planning load is high, as speakers allocate cognitive resources to manage the disruption (Zayats and Ostendorf, 2019; Székely et al., 2017).

Indeed, for repetitions, only the frequency of the following word and phonological complexity aligned with our predictions (see Section 2.3). Notably, the integration cost of the following word decreased repetitions, whereas a higher bigram probability of the following word increased repetitions, contradicting the prediction that lower predictability triggers disruptions (Hale, 2001; Sen, 2020; Arnold et al., 2007).

As mentioned earlier, although some properties of the preceding words guided the distribution of hesitations and repetitions, the properties of the following word influenced both phenomena. These findings align with models of incremental sentence production (Kempen and Hoenkamp, 1987), in which lexical access, syntactic integration, and phonological encoding and planning impose constraints on sentence processing and, in turn, its fluency (Fraundorf and Watson, 2013; Lickley, 2017).

Overall, our study extends previous work on speech disfluencies by showing that hesitations and repetitions arise from processing difficulties at distinct stages of the production pipeline and are selectively sensitive to different predictors depending on word position (Fraundorf and Watson, 2013; Lickley, 2017). Cross-linguistic evidence from Hindi

[8] Thanks to the anonymous reviewers for this suggestion.

highlights the need to move beyond head-initial languages in disfluency research, as typological organization fundamentally shapes how and where disruptions emerge. Future work should extend this line of inquiry to other head-final and typologically diverse languages to establish the generalizability of these findings.

Conclusion

This work presents a comprehensive investigation of how lexical, syntactic, phonological, and prosodic factors collectively predict disfluencies in Hindi. Our results reveal both similarities as well as dissimilarities between hesitations and repetitions. For example, factors such as phonological complexity robustly predict both these disfluencies. On the other hand, while hesitations are sensitive to cumulative processing pressures across both preceding and following words, repetitions are driven primarily by properties of the following word. These results show that hesitations and repetitions differ in terms of their determinants and locus. Our research highlights how the processes of incrementality/planning and monitoring in an SOV language like Hindi could be different from those in an SVO language like English (cf. Zafar, 2025). Future research should examine the universality of the observed patterns across typologically diverse languages to deepen our understanding of the production process cross-linguistically.

References

Jennifer E. Arnold, Michael Fagnano, and Michael K. Tanenhaus. 2003. Disfluencies signal thee, um, new information. *Journal of Psycholinguistic Research*, 32(1):25–36.

Jennifer E. Arnold, Carla L. Hudson Kam, and Michael K. Tanenhaus. 2007. If you say thee uh-you are describing something hard: The on-line attribution of disfluency during reference comprehension. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 33(5):914–930.

Dale J Barr, Roger Levy, Christoph Scheepers, and Harry J Tily. 2013. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of memory and language*, 68(3):255–278.

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.

Alan Bell, Jason M. Brenier, Michelle Gregory, Cynthia Girand, and Dan Jurafsky. 2009. Predictability effects on durations of content and function words in conversational english. *Journal of Memory and Language*, 60(1):92–111.

Elizabeth R. Blackmer and Janet L. Mitton. 1991. Theories of monitoring and the timing of repairs in spontaneous speech. *Cognition*, 39(3):173–194.

Kathryn Bock and Willem J. M. Levelt. 1994. Language production: Grammatical encoding. In *Handbook of Psycholinguistics*. Academic Press.

Weenink Boersma. 2012. Boersma, paul; weenink, david. *Praat: Doing phonetics by computer*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Heather Bortfeld, Silvia D. Leon, Judith E. Bloom, Michael F. Schober, and Susan E. Brennan. 2001. Disfluency rates in conversation: Effects of age, relationship, topic, role, and gender. *Language and Speech*, 44(2):123–147.

Susan E Brennan and 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.

Sabine Buchholz and Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the tenth conference on computational natural language learning (CoNLL-X)*, pages 149–164.

Alexandra Canavan and George Zipperlen. 1996. CALLFRIEND Hindi. Web download. LDC Catalog No. LDC96S52.

Chiara Celata. 2020. Bottom-up probabilistic information in visual word recognition: interactions with phonological and morphological functions. *Language Sciences*, 78:101267.

Herbert H. Clark and Jean E. Fox Tree. 2002. Using uh and um in spontaneous speaking. *Cognition*, 84(1):73–111.

Francesca Cossavella and Javier Cevasco. 2021. The importance of studying the role of filled pauses in the construction of a coherent representation of spontaneous spoken discourse. *Journal of Cognitive Psychology*, 33(2):172–186.

Sumanth Dammalapati, Rajakrishnan Rajkumar, Saurabh Ranjan, and Saurabh Agarwal. 2021. Effects of duration, locality, and surprisal in speech disfluency prediction in english spontaneous speech. *Society for Computation in Linguistics*, 4(1):91–101.

Fernanda Ferreira. 1993. Creation of prosody during sentence production. *Psychological Review*, 100(2):233–253.

Jean E. Fox Tree. 1995. The effects of false starts and repetitions on the processing of subsequent words in spontaneous speech. *Journal of Memory and Language*, 34(6):709–738.

Jean E Fox Tree. 2001. Listeners' uses of um and uh in speech comprehension. *Memory & cognition*, 29(2):320–326.

Scott H. Fraundorf and Duane G. Watson. 2013. Alice's adventures in um-derland: Psycholinguistic sources of variation in disfluency production. *Language, Cognition and Neuroscience*, 29(9):1083–1096.

Victoria Fromkin. 1973. *Speech errors as linguistic evidence*, volume 77. Walter de Gruyter.

Victoria A Fromkin. 1971. The non-anomalous nature of anomalous utterances. *Language*, pages 27–52.

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341.

Merrill F. Garrett. 1975. The analysis of sentence production. In *Psychology of Learning and Motivation*, volume 9, pages 133–177. Elsevier.

Edward Gibson and 1 others. 2000. The dependency locality theory: A distance-based theory of linguistic complexity. *Image, language, brain*, 2000:95–126.

Frieda Goldman-Eisler. 1961. A comparative study of two hesitation phenomena. *Language and Speech*, 4(1):18–26.

Adam Goodkind and Klinton Bicknell. 2018. Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the 8th workshop on cognitive modeling and computational linguistics (CMCL 2018)*, pages 10–18.

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*.

Charles F Hockett. 1967. Where the tongue slips, there slip i. *To honor Roman Jakobson*, 2:910–936.

Daniel Jurafsky and James H Martin. 2013. *Speech and Language Processing: Pearson New International Edition PDF eBook*. Pearson Higher Ed.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul NC, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. 2020. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of the association for computational linguistics: EMNLP 2020*, pages 4948–4961.

Gerard Kempen and Eduard Hoenkamp. 1987. An incremental procedural grammar for sentence formulation. *Cogn. Sci.*, 11:201–258.

Anoop Kunchukuttan, Divyanshu Kakwani, Satish Golla, Avik Bhattacharyya, Mitesh M Khapra, Pratyush Kumar, and 1 others. 2020. Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages. *arXiv preprint arXiv:2005.00085*.

Willem J. M. Levelt. 1983. Monitoring and self-repair in speech. *Cognition*, 14(1):41–104.

Willem J. M. Levelt. 1989. *Speaking: From Intention to Articulation*. MIT Press.

Willem J. M. Levelt and Anne Cutler. 1983. Prosodic marking in speech repair. *Journal of Semantics*, 2(2):205–218.

Willem J. M. Levelt, Ardi Roelofs, and Antje S. Meyer. 1999. A theory of lexical access in speech production. *Behavioral and Brain Sciences*, 22(1):1–38.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Robin Lickley. 2017. Disfluency in typical and stuttered speech. *Book series Studi AISV*, 3:373–387.

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

Sibout Govert Nooteboom. 1980. Speaking and unspeaking: Detection and correction of phonological and lexical errors in spontaneous speech. In *Errors in linguistic performance: Slips of the tongue, ear, pen and hand/ed. By Victoria A. Fromkin*, pages 87–95. Academic Press Inc.

Byung-Doh Oh and William Schuler. 2023. Transformer-based language model surprisal predicts human reading times best with about two billion training tokens. In *Findings of the association for computational linguistics: EMNLP 2023*, pages 1915–1921.

Benu Pareek, Mudafia Zafar, Meghna Hooda, Karan Yadav, Ashwini Vaidya, and Samar Husain. 2025. Iit delhi dialogue corpus: a quantitative analysis of a spoken corpus of hindi. *Language Resources and Evaluation*, pages 1–36.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and 1 others. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Albert Postma and Herman Kolk. 1993. The covert repair hypothesis: Prearticulatory repair processes in normal and stuttered disfluencies. *Journal of Speech and Hearing Research*, 36(3):472–487.

R Core Team. 2025. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Sidharth Ranjan, Rajakrishnan Rajkumar, Sumeet Agarwal, and IISER Bhopal. 2020. Forward surprisal models production planning in reading aloud. In *Proceedings of the 26th Architectures and Mechanisms for Language Processing Conference (AMLaP), Potsdam, Germany. University of Potsdam*.

Ardi Roelofs. 1992. A spreading-activation theory of lemma retrieval in speaking. *Cognition*, 42(1–3):107–142.

Douglas Roland, Hongoak Yun, Jean-Pierre Koenig, and Gail Mauner. 2012. Semantic similarity, predictability, and models of sentence processing. *Cognition*, 122(3):267–279.

Gregory Scontras, William Badecker, Lisa Shank, Eunice Lim, and Evelina Fedorenko. 2015. Syntactic complexity effects in sentence production. *Cognitive science*, 39(3):559–583.

Priyanka Sen. 2020. Speech disfluencies occur at higher perplexities. In *Proceedings of the Workshop on the Cognitive Aspects of the Lexicon*, pages 92–97. Association for Computational Linguistics.

Elizabeth Ellen Shriberg. 1994. Preliminaries to a theory of speech disfluencies. *Doctoral dissertation, University of California at Berkeley*.

Nathaniel J Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Éva Székely, Joseph Mendelson, and Joakim Gustafson. 2017. Synthesising uncertainty: The interplay of vocal effort and hesitation disfluencies. In *Proceedings of INTERSPEECH*, pages 804–808.

Walter JB Van Heuven, Pawel Mandera, Emmanuel Keuleers, and Marc Brysbaert. 2014. Subtlex-uk: A new and improved word frequency database for british english. *Quarterly journal of experimental psychology*, 67(6):1176–1190.

Michael S Vitevitch, Jonna Armbrüster, and Shinying Chu. 2004. Sublexical and lexical representations in speech production: effects of phonotactic probability and onset density. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 30(2):514.

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.

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

Bodo Winter. 2019. *Statistics for linguists: An introduction using R*. Routledge.

Himanshu Yadav, Shubham Mittal, and Samar Husain. 2022. A reappraisal of dependency length minimization as a linguistic universal. *Open Mind: Discoveries in Cognitive Science*, 6:147–168.

Mudafia Zafar. 2025. Planning scope in production: How grammar shapes incrementality. *Doctoral dissertation, Indian Institute of Technology Delhi*.

Vicky Zayats and Mari Ostendorf. 2019. Giving attention to the unexpected: Using prosody innovations in disfluency detection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 86–95, Minneapolis, Minnesota. Association for Computational Linguistics.

A Appendix

A.1 Linguistic feature computation

The procedure used to compute each linguistic feature is described below.

Word Frequency Raw token counts were tallied across the entire corpus using Python's `Counter`. These counts were converted to the Zipf scale using the formula:

$$\text{Zipf}(w) = \log_{10}\left(\frac{f_w}{N} \times 10^9\right) + 3$$

where f_w is the raw count of the word w and N is the total number of tokens in the corpus (Van Heuven et al., 2014).

Integration Cost Integration cost was computed as the linear distance between a syntactic head and its dependent, extracted directly from the dependency-annotated structures provided in the corpus (Gibson et al., 2000; Futrell et al., 2015).

Surprisal IndicBERT (Kakwani et al., 2020) (`ai4bharat/indic-bert`), a masked language model pre-trained on Hindi and other Indian languages, was used to compute surprisal (Smith and Levy, 2013; Goodkind and Bicknell, 2018; Wilcox et al., 2023; Oh and Schuler, 2023). The context was derived by joining all tokens preceding the target word.[9] The context of the following word did not include the hesitation token, as it may not

[9]We note that with regard to surprisal computation, Wilcox et al. (2023) found that multilingual language models perform similar to monolingual language models. In addition, their work shows that BERT-based masked models perform as well as autoregressive models. Finally, in our study, the context for surprisal calculation was limited to already spoken words, thus ensuring for the psychological validity of the surprisal values (cf. Wilcox et al., 2023).

have a lexical-level representation in the trained model, and its inclusion would confound the probability estimates. The context and target word were passed through the model to extract logits. The softmax function was applied to the logits to obtain the probability of the target token given its context. Surprisal was then computed as:

$$S(w_{n-1}) = -\log P(w_{n-1} \mid w_1, \ldots, w_{n-2})$$

$$S(w_{n+1}) = -\log P(w_{n+1} \mid w_1, \ldots, w_{n-1})$$

Forward Surprisal Forward surprisal was computed based on the same principles as surprisal stated above, the distinction being that the context was limited to unspoken upcoming words (Ranjan et al., 2020).

Semantic Similarity Hindi FastText vectors (`cc.hi.300.bin`) (Bojanowski et al., 2017) were used to extract vector embeddings. The context representation was the mean of the embeddings of all words to the left of the target position. Similar to surprisal, hesitation tokens were excluded from the context of the following word. Cosine similarity was computed between this mean vector and the target word's embedding using `scikit-learn` (Pedregosa et al., 2011).

Bigram Probability Unigram and bigram counts were collected from all sentences in the corpus. Bigram probability was computed as $C(w_{n-2}, w_{n-1})/C(w_{n-2})$. When a bigram was unavailable, for instance, when the hesitation appeared in the second position in the sentence and no two preceding words existed, a unigram fallback $C(w_{n-1})/N$ was applied for the preceding word and $C(w_{n+1})/N$ for the following word, where N is the total number of tokens in the corpus.

Word Length Word length was measured as the number of syllables in the target word, obtained using the `indicnlp` orthographic syllabifier for Hindi (Kunchukuttan et al., 2020).

Phonological Complexity Words were syllabified using the `indicnlp` orthographic syllabifier for Hindi (Kunchukuttan et al., 2020), such that syllables typically corresponded to consonant-vowel (CV) sequences. Phonological complexity was computed as the product of unigram and bigram syllable probabilities across the word. Since the first syllable has no preceding syllable, its probability was estimated as a unigram frequency in the full corpus. For all subsequent syllables, bigram transition probabilities were used, conditioned on the preceding syllable within the corpus.

A.2 Additional plots

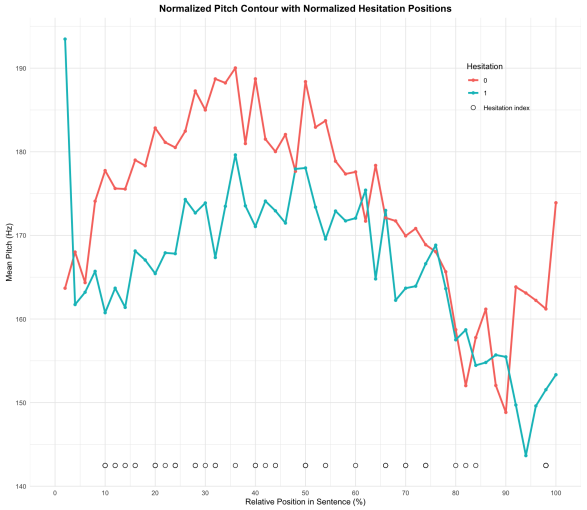


Figure 6: Normalized pitch contours for sentences with (red) and without (blue) hesitations. Hesitation sites show markedly lower pitch.

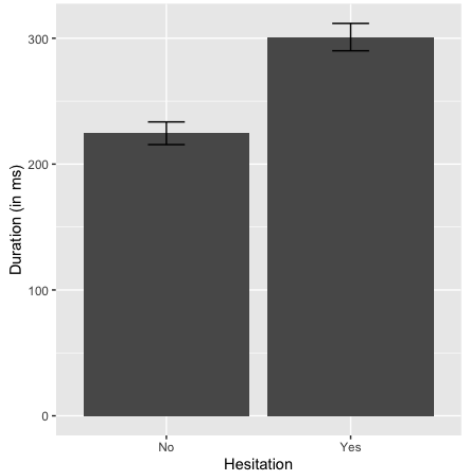


Figure 7: Mean duration of filled pauses and their corresponding no-hesitation tokens

Predictor	Hesitation			
	Coef	SE	z	p
Preceding word frequency	-2.87	1.29	-2.22	0.02603*
Following word frequency	-0.48	1.05	-0.45	0.64976
Preceding integration cost	2.80	1.13	2.48	0.01320*
Following integration cost	0.98	0.39	2.49	0.01281*
Preceding surprisal	1.97	0.79	2.47	0.01339*
Following surprisal	-6.46	3.32	-1.94	0.05182.
Following forward surprisal	5.91	3.13	1.88	0.05942.
Preceding semantic similarity	1.61	1.17	1.38	0.16791
Following semantic similarity	3.53	1.40	2.51	0.01207*
Preceding bigram probability	-0.62	0.45	-1.37	0.16981
Following bigram probability	-1.29	0.75	-1.71	0.08716.
Preceding word length	1.70	1.15	1.48	0.13948
Following word length	4.53	1.55	3.12	0.00181**
Preceding phonological complexity	11.60	3.96	2.93	0.00337**
Following phonological complexity	9.11	2.99	3.04	0.00236**
Preceding word frequency × word length	-2.52	1.10	-2.27	0.02280*
Preceding word frequency × Following forward surprisal	-0.25	0.58	-0.42	0.67060

*p<0.05, **p<0.01, **p<0.001

Table 3: Fixed effects from *glmer* models predicting hesitations. Shaded rows indicate statistically significant effects.

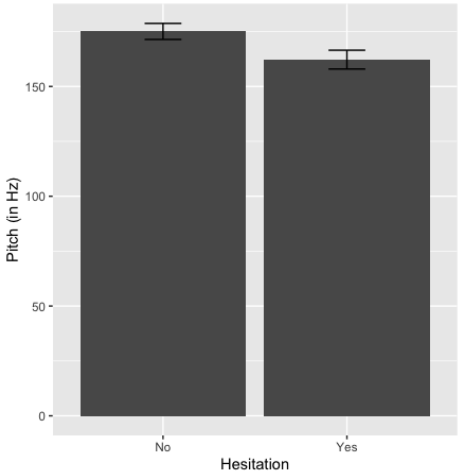


Figure 8: Mean pitch of filled pauses and their corresponding no-hesitation tokens

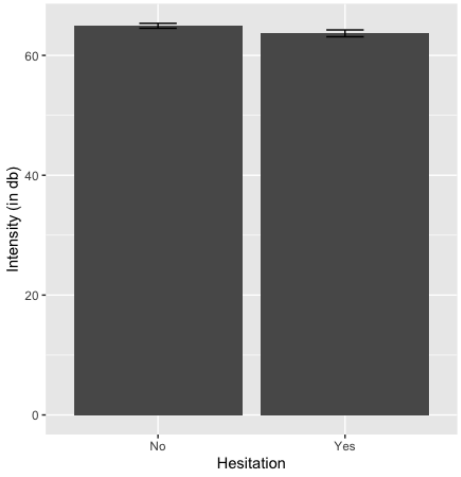


Figure 9: Mean intensity of filled pauses and their corresponding no-hesitation tokens

An LLM Investigation into Inherent and Structural Case Representation: a German Case Study

Iona Carslaw[1, 2], András Bárány[2], Itamar Kastner[2], Mark Steedman[1]

[1]School of Informatics, University of Edinburgh
[2]School of Philosophy, Psychology & Language Sciences, University of Edinburgh
i.c.a.carslaw@sms.ed.ac.uk {andras.barany, itamar.kastner, m.steedman}@ed.ac.uk

Abstract

A question for computational linguistics has been to what degree do language models encode case information. However, the majority of the work has focused on structural cases (cases which change when the syntactic configuration changes). On the other hand, inherent cases (which are assigned by specific lexical items and do not change if the syntactic configuration changes) have been overlooked. This paper sets out to investigate if German language models distinctly encode inherent dative from structural accusative and nominative. We conducted a linguistic probing investigation where probes are trained on contextual word embeddings of active nominative, accusative, and dative arguments to predict if passivised datives are analysed as a structural nominative. We provide a cased and caseless version of the experiment. Our results suggest that when case information is removed language models can distinguish between inherent dative and structural accusative, regardless of argument position, due to verb information. However, language models cannot distinguish between structural nominative and inherent dative when the dative appears in a position where there is an expected nominative, due to over-relying on surface patterns.

1 Introduction

A large amount of literature has been concerned with testing the syntactic abilities of language models as a means to understand neural net architectures and what (if anything) this can say about languages and how we learn them (Wilcox et al., 2018; Marvin and Linzen, 2018; Futrell et al., 2019; Papadimitriou et al., 2021; Kamerath and Santo, 2025; Kennedy, 2025, a.o.). A question amongst this work is to what degree do language models encode case information. Case here is defined as how DPs are morphologically marked to indicate their role in a sentence (Baker, 2015). Consider the active German sentence in 1a with the subject receiving nominative, and the object receiving accusative. If we passivise 1a, the case assignment changes with the new subject receiving nominative (1b[1]). Thus, a change in the syntactic configuration and the argument's syntactic position leads to a change in case assignment. These case types (accusative and nominative for German, but in other languages ergative and absolutive) are called *structural cases*.

(1)

a. *Der* *Metallurge* *hat* *den* *Cricketschläger* *geklaut*
the.NOM metallurgist.NOM has the.ACC cricket-bat.ACC stolen

'The metallurgist stole the cricket bat'

b. *Der* *Cricketschläger* *wurde* *geklaut*
the.NOM cricket-bat.NOM PASS.AUX stolen

'The cricket bat was stolen' (McFadden, 2024, p.177)

Given that a particular question is how language models encode arguments and what information is used to distinguish them (Papadimitriou et al., 2022; Mahowald et al., 2023; Lee et al., 2024; Wilson et al., 2023; Conia and Navigli, 2022), understanding how language models represent structural cases has become a research area (Papadimitriou et al., 2021; Mahowald et al., 2023; Ozaki et al., 2025). However, a language's case system can consist of more than just structural cases. German has dative which can be assigned by specific predicates to their objects (2a). However, unlike the structural cases, when the dative object gets passivised, the dative remains despite the change in syntactic configuration (2b). These types of cases are called *inherent cases*. Their representation within language models has yet to be investigated.

(2)

a. *Der* *Metallurge* *hat* *dem* *Dekan* *nicht*
the.NOM metallurgist.NOM had the.DAT dean.DAT not
gehorcht
obeyed

'The metallurgst didn't obey the dean'

[1]Glosses have been adapted.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 72–89
July 3-4, 2026 ©2026 Association for Computational Linguistics

b. *Dem* *Dekan* *ist* *nicht* *gehorcht* *worden*
the.DAT dean.DAT is not obeyed PASS.AUX

'The dean was not obeyed' (McFadden, 2024, p.177)

Past research has shown that language models favour utilising word-order strategies over morphological strategies (Czarnowska, 2022), and struggle generalising morphological information (Temesgen et al., 2025; Ismayilzada et al., 2025). Given the different behaviours of inherent and structural case, they provide a fruitful testing ground for how much morphological information (i.e. case) is being used by language models versus just word-order/surface patterns when encoding arguments. For instance, a German language model when generalising a structural case system can generalise that subjects receive nominative and objects receive accusative, and therefore just rely on syntactic positions. However, to predict the inherent dative, a model cannot just rely on syntax. Thus, this paper sets out to investigate how language models distinguish German structural and inherent cases from one another, by answering the following two questions: (i) are arguments which receive inherent cases versus those that receive structural cases distinguished from one another in the language model's representation; and (ii) do models default to a structural representation of arguments instead of a case representation. Our results suggest that German language models distinguish between structural accusative and inherent dative due to verb information, but when it comes to structural nominative and inherent dative, the models rely on surface patterns[2]

2 Background

Previous work shows that language models encode structural case information. Papadimitriou et al. (2021) conducted a probing investigation to see if language models can learn high-order structural case information. They found that *mBERT* represents arguments differently depending on how a language's structural cases are organised. Mahowald et al. (2023) questioned if language models can predict subject and objects when word order information is not present. They found that language models trained on cased languages performed better than those that do not. Although, they do state that the margin of improvement is minimal. Ozaki et al. (2025) found that a Hindi-Urdu LSTM had a linear encoding of structural cases which the model used in computing agreement tokens. However, doubts have been raised about the ability of language models to generalise morphological information. Czarnowska (2022) found that neural parsers have a tendency to over-rely on word order and lexical semantics, even if the language's primary way of indicating syntactic relations is through morphology. Weissweiler et al. (2023) conducted the famous Wug test (Berko, 1958) on *ChatGPT* and found that it fell short from human achievement and models that are trained for morphological analysis. Haley (2020) showed similar results with *BERT*. Many other studies have also raised doubts that language models have an abstract representation of morphological rules - inflectional or derivational (Ismayilzada et al., 2025; Temesgen et al., 2025; Weller-Di Marco and Fraser, 2024). Given there is evidence that language models encode structural case, but do not generalise morphological rules, we should investigate instances where case information diverges from structural positioning (i.e. inherent case).

[2]Part of the code is available at: https://github.com/ionacarslaw/German_Inherent_Structural_Probe

Train		**Test**
Label: 0	**Label: 1**	
1: Nom + Acc	Dat	Dat
2: Nom	Dat	Dat
3: Acc	Dat	Dat
4: Nom	Dat + Acc	Dat
5: Acc	Dat + Nom	Dat

Table 1: Different data configurations for the binary training classes.

3 Experimental Design

3.1 Probing Design

To investigate how inherent and structural cases are represented in German language models, we undertook a probing experiment. We followed closely to the methodology in Papadimitriou et al. (2021), where we take contextual word embeddings of arguments to train classifiers. We are interested in answering the following: (i) if structural and inherent cases are meaningfully distinguished from one another in the language model's representation, and (ii) if, instead of a case representation, models default to a structural/word-order representation of arguments. To test this, we used a training set of active structural nominatives (1a), structural accusatives (1a), and inherent datives (2a). Our test set consisted of passivised inherent datives (2b).

The main task in answering (ii) is, given class 0 of active structural nominatives and accusatives, and class 1 of active inherent datives, what would a probe classify passivised inherent datives as? If a model has just structural case representation and relies purely on structural argument encoding, then the passivised inherent argument will be classified alongside the nominatives/structural cases (class 0), if the models do not just use structural sequences and correctly model inherent case behaviour, then it will be classified alongside the other inherent datives (class 1). This set-up also helps answer (i) with the classes being along an inherent-structural case split. However, to fully answer if an inherent-structural dimension is a meaningful dimension that language models encode case, we include other data configurations, as it may be that a language model does not specifically distinguish between inherent and structural cases, but nonetheless the probe could extrapolate that information through other means. Therefore, we include a data configuration where nominative is class 0 and dative and accusative are class 1, and another where accusative is class 0 and nominative and dative are class 1 (see configuration 4 and 5 in Table 1). If the models meaningfully distinguish inherent and structural cases, the inherent and structural split (configuration 1) will be the most accurate. We have also included configurations where there are only two cases to see how much the classifier is training on subjecthood and objecthood (configuration 2 and 3).

3.2 Argument Selection

In German, structural nominatives and accusatives are the canonical case assignment, therefore specific criteria is not needed for them to arise. However, inherent cases are distinct in that they need certain lexical items to assign them. There are three distinct verbs groups in German which assign an inherent dative[3]; malefactive/benefactive verbs, where the malefactive/benefactive object is dative (2a), ditransitives, where the recipient object is dative (3a), and experiencer-stimulus verbs, where the experiencer object is dative (3b). These verbs differ in their argument structure (two versus three arguments), their theta roles (benefactive/malefactive, recipient, experiencer), and what syntactic operations they can undertake (e.g. experiencer-stimulus verbs cannot undergo passivisation).

(3)

a. *Er* *gibt* *seinem* *Meerschwein* *Futter*
he.NOM gives his.DAT guinea.pig.DAT food.ACC

'He gives food to the guinea pig'

b. *Der* *Tänzerin* *gefällt* *die* *Musik*
the.DAT dancer.DAT pleases the.NOM music.NOM

'The music pleases the dancer' (Haspelmath and Baumann, 2013)

A case system requires generalising high-order information (Papadimitriou et al., 2021). Therefore, to actually test the models case-generalising ability, we have made sure that the inherent datives in the training set are distinct from the test set. Therefore, our training set datives consisted of active datives from ditransitives and experiencer-stimulus verbs. Our test set consisted of passivised datives from malefactive/benefactive verbs (2b). Setting up the test set as so means we can determine if the probe has merely learnt that datives are a class due to them occurring with ditransitives and experiencer-stimulus verbs; if so the error rating on the test set will be around chance as how the dative are classified will not generalise over to malefactive/benefactive verbs.

4 Experiment 1: Cased Representation

4.1 Datasets

To build the datasets, we used the German UD corpora (McDonald et al., 2013) and native speakers. From UD, we removed any sentences that had sentence or argument coordination by looking for `ccomp`, and any with subordinate clauses by looking for `xcomp`. To gather the nominatives, we searched for `nsubj` and checked if `Case=Nom`. If the relationship was `nsubj:pass`, or the argument was a proper noun (`PROPN`) - which in German does not exhibit overt case - we skipped it. All other instances were added to the nominative dataset. For the accusatives, we searched for `obj` and checked if `Case=Acc`. If the argument was a proper noun we skipped it, otherwise all others were added to the accusative set. To gather the datives, we searched for sentences which contained verbs which assign dative. If an argument had `obj`

[3]German has inherent genitive and (arguably) inherent accusative. For the inherent genitive, we decided to not include them in our experiment, specifically because German genitive can naturally occur in a subject due to non-inherent case assignment means. For the inherent accusative, it is argued to be in the following: *mich.ACC friert* 'I'm freezing' (McFadden, 2008, p.64). However, *mich* cannot undergo A-movement (of which passivisation is an instance of); because we diagnose inherent cases as those that survive under A-movement, we do not count these as inherent. Both of these instances were removed from the data set.

or obl:arg and Case=Dat, they were set as potential training instances. If an argument had nsubj or nsubj:pass and Case=Dat they were set aside for potential training instances. Two native speakers went through the instances of datives and checked for the training set if: (i) these were instances of inherent datives, (ii) they were objects, and (iii) these datives were assigned by ditransitives or the experiencer-stimulus verbs[4]. For the test data, the native speakers checked if: (i) these were instances of datives, (ii) they were passivised, and (iii) these datives were assigned from malefactive/benefactive verbs. If they found an instance of a dative object assigned by a malefactive/benefactive verb, they passivised the sentence and added the entry to the test set. This process meant there were 39 instances of passivised inherent datives and 198 instances of active inherent datives. To increase the training and test size of the dative, the native speakers created new sentences. They had access to the nominative and accusative instances to emulate the subject areas and sentence structures the UD corpora has. For all the tasks involving the annotators, both checked the others work and if they disagreed on a judgement then they removed those data instances from the datasets. The total amount of training datives (including the UD instances) was 2021, and the total amount of test datives (including the UD instances) was 144. To make sure there were no class imbalances, a random selection of the nominatives and accusative occurred so that the training set had 2021 instances of accusatives and 2021 instances of nominatives. For data configurations where a single class was made up of two cases, 50% of each case was randomly selected so there was not a class imbalance. Approx 10% of the training set was withheld as a validation set so after the classifiers were trained, the layers that were investigated were chosen on accuracy of the validation set rather than the test set.

Control Datasets Given that the data has come from two different sources, we created control datasets to see if dataset characteristics played a role in the classifier accuracy. Alongside the randomly selected conditions, we built a *noun only* dataset (1234 training instances per case of which ∼10% for validation, 192 dative test instances), a *pronoun only* dataset (214 training instances per case of which ∼10% for validation, 52 dative test instances), and a dataset where the overall sentence *token amount* per case had the same distribution (1336 training instances per case of which ∼10% for validation, 244 dative test instances). For the *same token amount* control dataset we tokenised sentences with the *mBERT* tokeniser, counted token per data entry for token amounts, and made sure the overall distribution of token amounts were the same across cases. We conducted a one-way ANOVA to confirm that there was not a difference between case and token amount for this control dataset, and it confirms that there is not a statistically significant difference ($F(2, 4005) = 0.31$, $p = 0.736$).

[4]For a full list of the datives predicates see appendix E.

4.2 Models and embeddings

BERT We obtained contextual word embeddings from *bert-base-german-cased* (Chan et al., 2019); a case-sensitive German only language model with a bidirectional transformer encoder architecture (Vaswani et al., 2017), with 12 hidden layers and 110 Million parametres. Hyperparametres were set at default. If an argument had multiple tokens, the token embeddings were averaged.

mBERT We obtained contextual word embeddings from *bert-base-multilingual-cased* (Devlin et al., 2018); a case-sensitive multilingual language model with a bidirectional transformer encoder architecture (Vaswani et al., 2017), with 12 hidden layers and 110 Million parametres. Hyperparametres were set at default. If an argument had multiple tokens, the token embeddings were averaged.

LLäMmlein 1B We obtained contextual word embeddings from *LSX-UniWue/LLaMmlein_1B* (Pfister et al., 2025); a German only language model based on the decoder-only transformer architecture (Vaswani et al., 2017), with 22 hidden layers and 1 Billion parametres. Hyperparametres were set at default. If an argument had multiple tokens, the token embeddings were averaged.

Word2Vec Baseline We used static word embeddings as a baseline because the embeddings do not represent the surrounding sentence context of the arguments. If the Word2Vec embeddings represents the different cases, it is then due to the distribution of the lexical items. We used a case-sensitive German only static model (Müller, 2015) which was trained with gensim's word2vec library. A skip-algorithm was implemented with a sliding window of 5. Words with a frequency lower than 50 were ignored. The total vocabulary size of the

model is 608,130. For noun phrases, the noun embedding and determiner embedding were averaged.

Given this model does not include a tokeniser, and relies on a word appearing in its training data, some arguments could not be embedded. To mitigate this, if a word did not have an embedding, we undertook three processes to see if one was available. The first was capitalising pronouns (ignoring pronouns which have a distinction with uppercase, e.g. sie/Sie). The second was removing accents in case a data-cleaned version of the word was in the vocabulary. The third concerned German's productive ability to create new words through compounding. Here we used the compound splitter *Charsplit* (Tuggener, 2016) on the nouns in our dataset; if the first suggested compound split had a probability above 0, we checked if the two suggested words had embedding representation. If so they were averaged together. All other instances were removed from the static training, this meant 267 datives (of which 35 were from the test set), 313 accusatives, and 299 nominatives were removed from the Word2Vec datasets.

4.3 Classifier

Our classifier was a neural network with a single hidden layer of size 64, trained for 20 epochs on our training data. The classifier was trained to predict 0 or 1, with 1 always representing the dative in every data configuration. A probe was trained for each model (*BERT*, *mBERT*, *LLäMmlein 1B*, *Word2Vec*), and for each model's layers, for every different data configuration (*Acc* $\sim$ *Dat*, *Nom* $\sim$ *Dat*, *Nom/Dat* $\sim$ *Acc*, *Nom/Acc* $\sim$ *Dat*, *Nom* $\sim$ *Acc/Dat*), and for every control dataset (*Random*, *Same Token Amount*, *Noun Only*, *Pronoun Only*). This meant that for *BERT* we trained 13 layers $\times$ 5 data configurations $\times$ 4 datasets = 260 classifiers; for *mBERT* we trained 13 layers $\times$ 5 data configurations $\times$ 4 datasets = 260 classifiers; for *LLäMmlein 1B* we trained 23 layers $\times$ 5 data configurations $\times$ 4 datasets = 460 classifiers; and for *Word2Vec* we trained 1 layer $\times$ 5 data configurations $\times$ 4 datasets = 20 classifiers. This totals 1000 classifiers.

4.4 Results

The error rates on the dative test sets are given in Figures 1 and 2. For each model, only the layer which achieved the most accurate classifications on the validation set in the *Nom/Acc* $\sim$ *Dat* configuration is shown; for *BERT* this is layer 11, for *mBERT* this is layer 10, and for *LLäMmlein 1B* this is layer 21[5]. Across the board, all the models performed well. Although German exhibits some case syncretism (where the same case morpheme is used for different cases, e.g. *euch* in German is a second person plural pronoun for accusative and dative), the majority of the case morphemes are distinct from one another (e.g. *ich/mich/mir* is first person singular for nominative, accusative, and dative respectively). It is likely that the classifier has trained on the existence of these specific case tokens rather than actual case representation[6]. Due

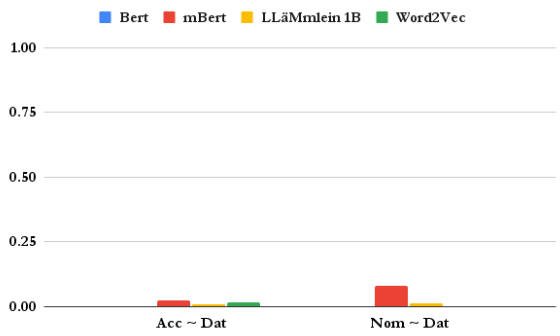


Figure 1: Error rate for the test set. Results show when the classifier trained on two cases.

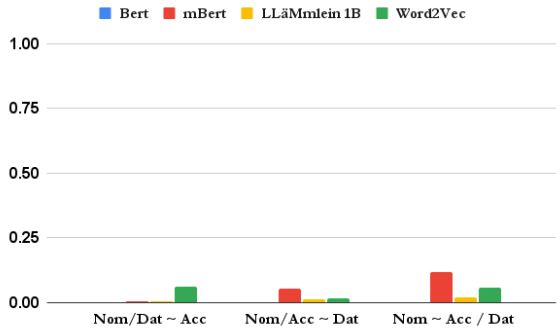


Figure 2: Error rate for the test set. Results show when the classifier trained on three cases.

[5]To see the full results, including the error ratings on the validation set and the control datasets, consult the appendix.

[6]An anonymous reviewer suggested that due to the embeddings coming from deep layers within all the models, specific token information would not be available for the classifiers to train on, and therefore an alternative explanation is needed for why they perform so well here. To play hypothetical, it could be that indeed a language model has encoded inherent cases, and the classifiers are picking up on such. Given the second experiment's results, that when you remove explicit case information, datives get analysed as a structural nominative, what could be argued is that the German language models have overgeneralised structural case assignment, in other words without a specific case signal the models default to a structural case representation as a strategy instead of what we claim to be overfitting to surface patterns. This is a testable claim; in child learners of German it has been noted that children

to this, we put these results to the side, and undertook a caseless version of this experiment, where we removed the case morphology.

5 Experiment 2: Caseless Representation

5.1 Caseless Datasets

We assume that the language models can handle caseless arguments given that proper nouns in German do not exhibit overt case marking. To create a caseless representation of the arguments, we removed case information and replaced it with a bundle of features. For pronouns, we had a bracket notation which included information such as gender (masculine, feminine, neuter), person (first, second, third), number (singular, plural), and pronoun type (personal, relative, interrogative, demonstrative). For instance, *wir/uns* (we/us - nominative and accusative/dative respectively) would become: `[Person=1Plural|Personalpronomen]`

For noun phrases, we lemmatised the noun to remove case declensions, and replaced the article with a bundle of features. The article features included article type (indefinite, definite, demonstrative, possessive), if it was a possessive article it included gender, person, number, and in general if the articles were plural that information was shown. However, although articles do agree in gender with the noun, because the noun was still represented, we did not include this information as it is still recoverable. For an example, *Seine Visionen* 'his visions' would become: `[Numerus=Plural|Geschlecht=Maskulin|Person=3Singular|Possessivartikel] Vision`

We achieved this by using SpaCy's morphological parser (Honnibal et al., 2020). If the predicted case from the parser did not match the case given, then it was done by hand.

5.2 Controls, Classifiers, and Models.

The control datasets *same token amount*, *noun only*, and *pronoun only* were kept the same, bar now having a caseless argument. The classifier architecture, the classifier training, and models used were all kept the same from the first experiment. However, with *Word2Vec*, given the lemmatising and the insertion of new words with the feature bundles, some of the words were not represented within its dictionary. Therefore, 492 datives (46 from the test set), 520 accusatives, and 511 nominatives were dropped from the *Word2Vec* datasets.

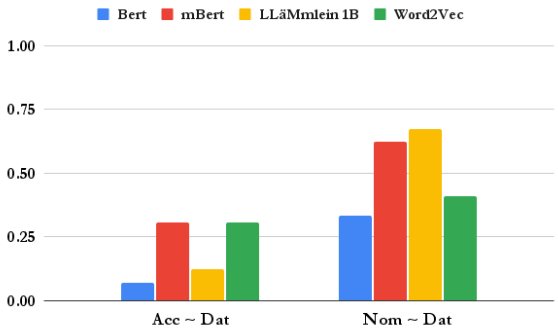


Figure 3: Error rates for the caseless test set. Results show when the classifier trained on two cases.

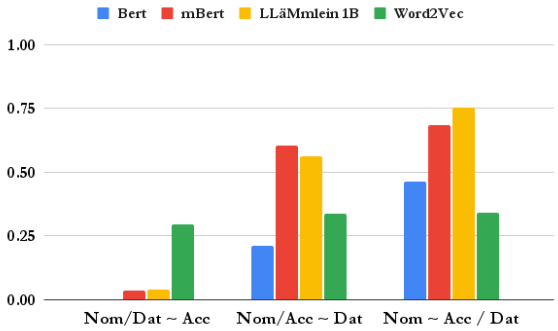


Figure 4: Error rates for the caseless test set. Results show when the classifier trained on three cases.

5.3 Results

The error rates on the caseless dative test sets are given in Figures 3 and 4. The layers shown are 11 for *BERT*, 10 for *mBERT*, and 21 for *LLäMmlein 1B*. For the data configurations that used only two cases (Figure 3), each model had a lower error rate with the *Acc* ∼ *Dat* probes, than the *Nom* ∼ *Dat* probes. Only *BERT* did better than the *Word2Vec* baseline in both configurations, with *mBERT* and *LLäMmlein 1B* performing worse than *Word2Vec* with the *Nom* ∼ *Dat* configuration. Concerning the control datasets, the same general trend of the *Acc* ∼ *Dat* probes performing better than the *Nom* ∼ *Dat* probes was seen, bar for *Word2Vec* in the *pronoun only* condition where both data configurations had the same error rate.

will overgeneralise the structural case system (nominative and accusative) rather than the dative, however in contrast Icelandic children overgeneralise the dative (see section 1.2 of Nowenstein (2023) for an overview). Given this, if it's true that the structural case system has been overgeneralised for the German language models, then we would expect the inherent case systems to overgeneralise for the Icelandic language models. If instead it's about surface patterns then we should expect the same results for both languages - although some controls would be needed for Icelandic given dative subjects can naturally occur.

For the data configurations that used three cases (Figure 4), all the models showed the same trends, where the data configuration with the lowest error rate was *Nom/Dat* ∼ *Acc*, and the data configuration with the highest error rate was *Nom* ∼ *Acc/Dat*. Concerning the control datasets, the same general trends were seen excluding the following exceptions: *mBERT* in the *pronoun only* condition had the same error rates for *Nom/Acc* ∼ *Dat* and *Nom* ∼ *Acc/Dat*; *LLäMmlein 1B* in the *pronoun only* condition had the highest error rate be *Nom/Acc* ∼ *Dat*; *Word2Vec* in the *noun only* and *pronoun only* condition had the highest error rate be *Nom/Acc* ∼ *Dat*.

6 Discussion

The NLP Take: *BERT* better distinguishes German inherent and structural cases when compared to *mBERT* and *LLäMmlein 1B*. This indicates that smaller unilingual models encode a more consistent inherent case representation. For why a unilingual model represents case better than a multilingual model, it could be that a consistent inherent case generalisation across languages is harder to learn. Consider the dative; a cross-linguistic definition can be described as indicating (i) low-transitivity and (ii) the affectedness/volition of an argument (Næss, 2008); this provides a good description of the German instances, which are used on experiencers, benefactives, and recipients. However, many languages which have dative utilise other cases for the supposed dative function; e.g. in Kalkatungu (an Australian indigenous language [Blake 1983]) dative represents benefactives, but allative represents recipients (Blake, 2001). The syntactic position of inherent datives can also differ between languages; e.g. German inherent datives occur on objects, whilst Icelandic can assign dative to subjects (Jónsson, 2003). Thus, dative case is not homogenous crosslinguistically - making it harder for multilingual models to generalise. Of course, as shown in Papadimitriou et al. (2021) multilingual models can generalise case information crosslingually, but crucially these were structural cases being probed, not inherent. Inherent cases require more information than just structural sequencing; they require language-specific lexical knowledge, by knowing which specific lexical items take an inherent case. This specific knowledge is truly to do with lexical items rather than more general semantic roles, given that there are German verbs which have the same semantic roles, but do not take inherent cases, e.g. *gratulieren* and *beglückwünschen* both mean 'to congratulate', but the former takes a dative object, and the latter accusative (McFadden, 2008); meaning a model cannot generalise inherent cases to semantic roles. Therefore, it may be that unilingual models better represent language-specific lexical requirements, compared to multilingual models.

For why a smaller language model represents case better than a larger model, this could be because the larger model has overfit more to surface syntactic patterns. A strategy to distinguish between structural accusative and inherent dative (regardless of if the dative is passivised) in the caseless condition is to train on which verbs are present in the larger context, as transitive verbs that assign an inherent dative and those that default to structural case (and therefore get accusative objects) are in a complementary distribution. Of course, given that the test datives used verbs which were distinct from the training dative verbs, for the classifier to achieve a low error rate on test set, the benefactive/malefactive verbs have to be encoded as being the same as the stimulus-experiencer verbs and ditransitives case-wise. In Figure 3 we can see that both *BERT* and *LLäMmlein 1B* are using such information given that they are below the static baseline in *Acc* ∼ *Dat*, where such context information is not available.

When we come to the *Nom* ∼ *Dat* classifiers however, verb information cannot be utilised because nominatives occur with predicates which select an inherent dative. For both *BERT* and *LläMmlein 1B* we now see a higher error rate than the *Acc* ∼ *Dat* condition, with *LläMmlein 1B* performing worse than the static baseline. This is in contrast with low error rates on the *Nom* ∼ *Dat* validation set (0 and 0.008 for *BERT* and *LläMmlein 1B* respectively). However, this can be explained if the language models are utilising a structural strategy combined with lexical knowledge to distinguish arguments. Consider that a way to split structural nominative, structural accusative, and inherent dative in German, is by encoding nominatives as what occurs on subject, datives as what occurs on objects when specific group of verbs X occurs, and accusatives as what occurs on objects

with all other verbs[7][8]. This strategy explains why for both *BERT* and *LläMmlein 1B* have in all conditions (including the three case configurations) low error rates on the validation sets. However, when the dative argument gets passivised, it gets realised as a subject and therefore patterns with the nominative. This is interesting given that there are distinct patterns between passivised inherent datives and passivised arguments which receive nominative; for instance German does not allow subject-verb agreement with passivised datives and instead opts for default agreement. Thus, the difference in performance from *BERT* and *LläMmlein 1B* is that *LläMmlein 1B* has overfit to surface patterns concerning passivised arguments, in that passivised inherent datives are passivised subjects, when *BERT* has less done so. However, this is not to give *BERT* too much credit given that in the *Nom* $\sim$ *Acc/Dat* condition it is worse than the static baseline and is basically at random chance, and therefore has also overfit, just not to the same degree.

This structural and lexical strategy of distinguishing cases can also explain the pattern for all the models in the three case conditions where they performed best in the *Nom/Dat* $\sim$ *Acc* condition and worst in *Nom* $\sim$ *Acc/Dat*, because in *Nom/Dat* $\sim$ *Acc* you can use the verb information and structural positioning to distinguish the arguments, in *Nom/Acc* $\sim$ *Dat* only the verb information, and in *Nom* $\sim$ *Acc/Dat* no such information. However, there is an interplay with the different distributions of pronouns and nouns per case as seen by the static baseline being better than random chance and due to the control datasets of *pronoun only* and *noun only* changing the error rates for *Nom/Acc* $\sim$ *Dat* and *Nom* $\sim$ *Acc/Dat* with *LläMmlein 1B* and *mBERT*. However, crucially, with the control datasets, the *Nom/Dat* $\sim$ *Acc* condition was always the best performing configuration. So although there is some argument type distribution per case, this is not the main information the models use to distinguish the case. Of course, to confirm this structural and lexical encoding of inherent and structural case information, a follow-up experiment should conduct a causal intervention experiment (Geiger et al., 2021).

The Linguistics Take: A recent minimalist theory in how to account for the distinct behaviours of inherent and structural case, has been to synthesise two competing case accounts (McFadden, 2024). We refer the reader to the paper for a full lay out of the synthesis, but in simple terms inherent cases have their own syntactic head (i.e. in the syntactic domain), whilst structural cases do not have their own syntactic head but instead are the morphemes that are inserted on bare DPs post-spellout (i.e. in the phonological domain). A prediction of such is that when encoding case information, inherent and structural cases should be distinctly represented because they are different phenomena.

Although the field is debating where language models should fall in theory creation (e.g. Chomsky et al. (2023) versus Piantadosi (2023)), we can agree that language models produce grammatical text and *seemingly* do not over-generate. Given this, if inherent and structural cases have such distinct behaviours/distributions, we may expect a language model to utilise this information when representing cased argument, and thus to distinctly represent inherent and structural cases. Our results show that this is not true, as the three-way case configuration with a distinction between inherent and structural cases did not have the most accurate probe. Although language models cannot prove any linguistic theory, they can at least show us if distributions are prevalent enough to learn from and if certain information is needed for a grammatical producing system. Here, the distribution between structural accusative and inherent dative is distinct and thus beneficial to encode, but the same cannot be said for the inherent dative and structural nominative. What we may want to take from this is that although structural and inherent cases have distinct behaviours they are not distinct enough to warrant being encoded by language models, and therefore we may not want to theoretically include such a strong distinction between the two case types.

To answer our questions: are arguments which receive inherent cases versus those that receive structural cases distinguished from one another? No, although the three cases can be distinguished from one another, there does not seem to be a specific encoding to distinguish inherent status from structural status. Do models default to a structural

[7]The hard part is figuring out what is the subject, especially given that German can have variable word order. However, language models have been shown to accurately predict subject and objects when word order information is removed and instead rely on lexical cues (Mahowald et al., 2023).

[8]To account for the ditransitive, you would need to add an extra stipulation that accusative occurs with all other verbs/objects which are left over with no case. This does raise an interesting question of how models deal with indirective or secundative marking on ditransitives - see Haspelmath (2005).

representation of arguments instead of a case representation? Yes, as passivised inherent datives are interpreted as a passive subject despite not having full subjecthood status. Overall, these results fall in line with Czarnowska (2022) in that models overrely on word patterns and lexical semantics despite a language's morphology, and raises doubts of how well language models generalise case information.

7 Conclusion

Previous investigations into how language models represent case has mainly focused on structural cases, with inherent cases being overlooked. However, inherent cases have distinct behaviours and require different generalisations to accurately represent them. By utilising linguistic probing, we found that when case tokens are removed, language models can distinguish between inherent dative and structural accusative, regardless of argument position, due to verb information. However, language models struggle to distinguish between structural nominative and inherent dative when the dative argument is passivised due to over-relying on surface patterns. This adds to a larger body of work which shows that language models do not utilise morphology information, but instead use syntactic sequences and word items to disambiguate language.

8 Limitations and Ethical Considerations

There are three limitations to this project: (i) the difficulty of expanding into other languages, (ii) the datasets coming from two different sources, and (iii) a lack of models with differing tokeniser strategies. In response to the first limitation, there are other languages which also have inherent cases (e.g. Hungarian, Polish, Icelandic). However, inherent cases can be particularly language-specific, with other cases than dative being used, or occurring on subjects instead of objects. It is also not guaranteed that the inherent case can be passivised (consider the German stimulus-experiencer verbs), and instead a different test construction would have to be used (e.g. some other A-movement construction). Given this, considerable work would need to be undertaken to expand to a different language. Not only this, but considering the frequency problem with these cases, it would be likely that anyone attempting a different language would face the same difficulty of UD not having a significant size of the specific cases/constructions needed. As a means to address how other languages may have different case representations, we included a multilingual model *mBERT* in our investigation which does include other languages with inherent cases. There is also the other side of this limitation which cannot be solved with multilingual models, in that there are many languages with case systems which do not have language models. To best mitigate this problem, we have made sure that our paper specifically mentions we work on German, and have also not overstated our claims to apply to all languages. Concerning the second limitation of how the majority of our dative training, validation, and test set comes from a different source to the accusative and nominative dataset; we dealt with this by including control datasets which tested for if different dataset characteristics played a role in the representation. For the third limitation concerning lack of differing tokeniser strategies, we do test sub-word tokenisation versus word tokenising (e.g. *Word2Vec*), however, given that *Word2Vec* creates a static embedding it is unclear how much of an impact the tokenising has had versus the static nature of the embedding. Simply, the experiment can be repeated, however, instead of changing model size and architecture, you would test along different tokenising strategies.

On the ethical considerations, it is always good to consider the broader ethical implications of using language models in the first place. Concerning this, we used a minimal set of models which still represented different conditions as a means to reduce computing power and resources, and thus carbon emissions. We also used models where the original authors were transparent in their architecture and training, and can be used without having to give user details to the companies/authors of said models. There is, as stated in the limitations, the ethical problem of only working on high-resource languages (e.g. German), and how the work is not reproducible on low-resource languages, which are also commonly minoritised languages. Unfortunately, this is a broader problem in computational linguistics, and how we have attempted mitigate this problem is to clearly state we are working on German and to not overstate our claims.

Acknowledgements

This work was supported by the UKRI Centre for Doctoral Training in Natural Language Processing, and Iona Carslaw was funded by the UKRI (refer-

ence: EP/S022481/1). We would like to thank Pia Pratscher and Theresia Michel for their work on the dataset and German judgements. We would also like to thank the three anonymous reviewers, the participants at the LELPGC25 at the University of Edinburgh (where the first experiment of this paper was presented), Ella Markham for reading an initial draft, and Frank Keller for insightful comments.

References

Matt Baker. 2015. *Case: Its Principles and its Parameters*. Cambridge University Press.

Jean Berko. 1958. The child's learning of english morphology. *Word*, 14(2-3):150–177.

Barry Blake. 1983. Structure and word order in kalkatungu: The anatomy of a flat language. *Australian Journal of Linguistics*, 3(2):143–175.

Barry Blake. 2001. *Case*, 2nd edition edition. Cambridge University Press.

Branden Chan, Timo Möller, Malte Pietsch, and Tanay Soni. 2019. German bert. Accessed on 12th January 2026.

Noam Chomsky, Ian Roberts, and Jeffrey Watumull. 2023. The false promise of chatgpt. *The New York Times*.

Simone Conia and Roberto Navigli. 2022. Probing for predicate argument structures in pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4622–4632.

Paula Czarnowska. 2022. *Morphological competence in neural natural langauge processing*. Ph.D. thesis, The University of Camrbidge.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Robert Dixon. 1994. *Ergativity*. Cambridge University Press.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. *Advances in neural information processing systems*, 34:9574–9586.

Coleman Haley. 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.

Martin Haspelmath. 2005. Argument marking in ditransitive alignment types. *Linguistic discovery*, 3(1):1–21.

Martin Haspelmath and Luisa Baumann. 2013. German (standard). In Iren Hartmann, Martin Haspelmath, and Bradley Taylor, editors, *Valency Patterns Leipzig*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

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.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Mete Ismayilzada, Defne Circi, Jonne Sälevä, Hale Sirin, Abdullatif Köksal, Bhuwan Dhingra, Antoine Bosselut, Duygu Ataman, and Lonneke Van Der Plas. 2025. Evaluating morphological compositional generalization in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1270–1305, Albuquerque, New Mexico. Association for Computational Linguistics.

Jóhannes Gísli Jónsson. 2003. Not so quirky: On subject case in icelandic. *New perspectives on case theory*, 156:127–163.

Michael Kamerath and Aniello De Santo. 2025. Do llms disambiguate italian relative clause attatchment. In *Society for Computation in Linguistics*, volume 8, page 33.

Mary Kennedy. 2025. Evidence of hierarchically-complex syntactic structure within bert's word representation. In *Society for Computation in Linguistics*, volume 8, page 18.

Eun-Kyoung Rosa Lee, Sathvik Nair, and Naomi Feldman. 2024. A psycholinguistic evaluation of language models' sensitivity to argument roles. *arXiv preprint arXiv:2410.16139*.

Kyle Mahowald, Evgeniia Diachek, Edward Gibson, Evelina Fedorenko, and Richard Futrell. 2023. Grammatical cues to subjecthood are redundant in a majority of simple clauses across languages. *Cognition*, 241:105543.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. *Preprint*, arXiv:1808.09031.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Clausia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97.

Thomas McFadden. 2008. German inherent datives and argument structure. In *Datives and other cases: Between argument structure and event structure*, pages 49–77. John Benjamins Publishing Company.

Thomas McFadden. 2024. A synthesis for the structural/inherent case distinction and its comparative and diarchronic consequences. In Christina Sevdali, Dionysios Mertyris, and Elena Anagnostopoulou, editors, *The Place of Case in Grammar*. Oxford Academic.

Andreas Müller. 2015. Analyse von Wort-Vektoren deutscher Textkorpora.

Iris Edda Nowenstein. 2023. *Building yourself a variable case system: The acquisition of Icelandic datives*. Ph.D. thesis.

Åshild Næss. 2008. Varieties of datives. In *The Oxford Handbook of Case*. Oxford Academic.

Satoru Ozaki, Rajesh Bhatt, and Brian W Dillon. 2025. A lstm language model learns hindi-urdu case-agreement interactions, and has a linear encoding of case. In *Proceedings of the Society for Computation in Linguistics 2025*, pages 64–73.

Isabel Papadimitriou, Ethan A Chi, Richard Futrell, and Kyle Mahowald. 2021. Deep subjecthood: Higher-order grammatical features in multilingual bert. *arXiv preprint arXiv:2101.11043*.

Isabel Papadimitriou, Richard Futrell, and Kyle Mahowald. 2022. When classifying grammatical role, bert doesn't care about word order... except when it matters. *arXiv preprint arXiv:2203.06204*.

Jan Pfister, Julia Wunderle, and Andreas Hotho. 2025. LLäMmlein: Transparent, compact and competitive German-only language models from scratch. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2227–2246, Vienna, Austria. Association for Computational Linguistics.

Steven T Piantadosi. 2023. Modern language models refute chomsky's approach to language. *From fieldwork to linguistic theory: A tribute to Dan Everett*, 15:353–414.

Roland Schäfer. 2015. Processing and querying large web corpora with the COW14 architecture. In *Proceedings of Challenges in the Management of Large Corpora 3 (CMLC-3)*, Lancaster. UCREL, IDS.

Roland Schäfer and Felix Bildhauer. 2012. Building large corpora from the web using a new efficient tool chain. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 486–493, Istanbul, Turkey. European Language Resources Association (ELRA).

Tsedeniya Kinfe Temesgen, Marion Di Marco, and Alexander Fraser. 2025. Extracting linguistic information from large language models: Syntactic relations and derivational knowledge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27210–27226, Suzhou, China. Association for Computational Linguistics.

Don Tuggener. 2016. Incremental coreference resolution for german. Master's thesis, The University of Zurich.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schuetze, Kemal Oflazer, and David Mortensen. 2023. Counting the bugs in ChatGPT's wugs: A multilingual investigation into the morphological capabilities of a large language model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6508–6524, Singapore. Association for Computational Linguistics.

Marion Weller-Di Marco and Alexander Fraser. 2024. Analyzing the understanding of morphologically complex words in large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1009–1020, Torino, Italia. ELRA and ICCL.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do rnn language models learn about filler-gap dependencies? *arXiv preprint arXiv:1809.00042*.

Michael Wilson, Jackson Petty, and Robert Frank. 2023. How abstract is linguistic generalization in large language models? experiments with argument structure. *Transactions of the Association for Computational Linguistics*, 11:1377–1395.

A Full Bert Results

				layers												
				0	1	2	3	4	5	6	7	8	9	10	11	12
Random	Cased	Acc0_Dat1	Val	0.015	0.0175	0.0125	0.01	0.0125	0.01	0.01	0.005	0.0025	0	0	0	0
			Test	0.0204918	0.0204918	0.01639344	0.01639344	0.00409836	0.00409836	0.01229508	0	0	0	0	0	0
		Nom0_Dat1	Val	0.0175	0.0075	0.01	0.0075	0.0025	0.0025	0	0	0	0	0	0	0
			Test	0.09016393	0.07786885	0.08196721	0.06967213	0.04098361	0.04098361	0.0204918	0.0204918	0	0.00409836	0	0	0
		Nom1_Acc0_Dat1	Val	0.12166667	0.11	0.07333333	0.075	0.06333333	0.06166667	0.05	0.02833333	0.01	0.00833333	0.01166667	0.01166667	0.015
			Test	0.02459016	0.0204918	0.0204918	0.01639344	0	0	0	0	0	0	0	0	0
		Nom0_Acc0_Dat1	Val	0.02166667	0.01166667	0.01333333	0.01333333	0.005	0.00666667	0.00666667	0.00333333	0.00166667	0	0.00166667	0	0.00166667
			Test	0.05737705	0.05327869	0.03688525	0.03688525	0.03278689	0.0204918	0.02459016	0.00409836	0	0.00409836	0.00409836	0	0
		Nom0_Acc1_Dat1	Val	0.12166667	0.11333333	0.08166667	0.06333333	0.05166667	0.04333333	0.035	0.02333333	0.00666667	0.00833333	0.005	0.00333333	0.005
			Test	0.1352459	0.0942623	0.12704918	0.09016393	0.07377049	0.04508197	0.04098361	0.03278689	0.00409836	0.00409836	0	0	0.00819672
		Acc0_Dat1	Val	0.1875	0.1325	0.1325	0.095	0.1025	0.1025	0.09	0.0775	0.06	0.0675	0.0625	0.035	0.0375
			Test	0.14344262	0.22131148	0.20491803	0.26639344	0.2295082	0.22540984	0.19262295	0.13934426	0.1557377	0.24590164	0.19672131	0.06967213	0.04918033
		Nom0_Dat1	Val	0.2275	0.19	0.1825	0.155	0.135	0.1275	0.11	0.0825	0.0575	0.06	0.0625	0.0375	0.0275
			Test	0.62295082	0.46721311	0.46721311	0.4795082	0.51229508	0.52459016	0.5204918	0.51229508	0.43032787	0.59836066	0.63114754	0.33196721	0.19672131
	Caseless	Nom1_Acc0_Dat1	Val	0.20166667	0.2	0.19333333	0.16166667	0.14	0.13666667	0.14	0.115	0.09833333	0.10833333	0.11166667	0.05333333	0.045
			Test	0.07377049	0.06967213	0.07377049	0.06967213	0.05737705	0.06147541	0.07377049	0.01639344	0.0204918	0.02868852	0.0204918	0	0
		Nom0_Acc0_Dat1	Val	0.235	0.185	0.16666667	0.15166667	0.14333333	0.11833333	0.125	0.10333333	0.06166667	0.08666667	0.07166667	0.03833333	0.04
			Test	0.46311475	0.43852459	0.4057377	0.44262295	0.44672131	0.4057377	0.43852459	0.40163934	0.4057377	0.65163934	0.60655738	0.21311475	0.19262295
		Nom0_Acc1_Dat1	Val	0.21166667	0.19666667	0.19333333	0.17	0.16833333	0.16333333	0.165	0.13	0.09833333	0.09	0.06666667	0.07	0.05333333
			Test	0.72540984	0.58606557	0.5942623	0.6352459	0.68852459	0.65163934	0.68442623	0.66393443	0.68852459	0.65163934	0.67622951	0.46311475	0.35245902
		Acc0_Dat1	Val	0.01712329	0.02054795	0.01369863	0.01027397	0.01712329	0.01369863	0.01369863	0.00684932	0.00684932	0.00342466	0.00342466	0.00342466	0.00342466
			Test	0.03688525	0.01639344	0.01229508	0.00819672	0.01229508	0.00409836	0.01229508	0	0	0	0	0	0
		Nom0_Dat1	Val	0.02054795	0.01027397	0.01027397	0.00684932	0.01027397	0.01027397	0.00342466	0.00342466	0	0	0	0	0.00342466
			Test	0.08606557	0.05737705	0.05327869	0.03688525	0.02868852	0.02868852	0.03278689	0.0204918	0.00409836	0.00409836	0	0	0
	Cased	Nom1_Acc0_Dat1	Val	0.10958904	0.11415525	0.08675799	0.07534247	0.06392694	0.05707763	0.05479452	0.02283105	0.00913242	0.00684932	0.00684932	0.00456621	0.00228311
			Test	0.01229508	0.01229508	0.00409836	0	0	0	0	0	0	0	0	0	0
		Nom0_Acc0_Dat1	Val	0.02283105	0.02054795	0.01598174	0.01826484	0.00684932	0.01141553	0.01369863	0.00684932	0.00456621	0.00684932	0.00684932	0.00456621	0.00684932
			Test	0.03688525	0.06557377	0.05327869	0.04918033	0.0204918	0.0204918	0.03278689	0.00409836	0.00409836	0.00409836	0	0.00409836	0
		Nom0_Acc1_Dat1	Val	0.11187215	0.10958904	0.08219178	0.06849315	0.05936073	0.06164384	0.05022831	0.02283105	0.00228311	0.00456621	0	0	0.00228311
			Test	0.10655738	0.12704918	0.15983607	0.1147541	0.09016393	0.10245902	0.09836066	0.04918033	0.00409836	0	0	0	0.00409836
Same Token Amount		Acc0_Dat1	Val	0.20547945	0.16780822	0.17808219	0.13356164	0.12671233	0.15753425	0.11643836	0.09589041	0.08219178	0.07191781	0.04452055	0.03082192	0.03767123
			Test	0.19262295	0.20901639	0.21721311	0.20491803	0.18852459	0.18032787	0.18852459	0.13934426	0.15983607	0.26639344	0.20491803	0.03278689	0.05327869
		Nom0_Dat1	Val	0.20547945	0.19178082	0.17123288	0.15068493	0.14726027	0.11986301	0.12328767	0.06849315	0.04109589	0.05136986	0.05479452	0.0239726	0.04452055
			Test	0.63114754	0.55327869	0.52868852	0.58606557	0.59016393	0.54508197	0.5204918	0.52868852	0.54098361	0.57377049	0.62295082	0.32786885	0.30737705
	Caseless	Nom1_Acc0_Dat1	Val	0.21004566	0.18949772	0.19863014	0.16438356	0.15296804	0.14840183	0.14840183	0.12328767	0.11187215	0.10958904	0.07990868	0.05022831	0.0456621
			Test	0.08196721	0.07786885	0.06557377	0.06557377	0.05737705	0.04918033	0.05327869	0.02868852	0.0204918	0.04918033	0.0204918	0.00409836	0
		Nom0_Acc0_Dat1	Val	0.23972603	0.22146119	0.20547945	0.17808219	0.14155251	0.14155251	0.15068493	0.1347032	0.05936073	0.0913242	0.10045662	0.04109589	0.03196347
			Test	0.47540984	0.43442623	0.44262295	0.4795082	0.52868852	0.43852459	0.4795082	0.42213115	0.45491803	0.66393443	0.61065574	0.26229508	0.19262295
		Nom0_Acc1_Dat1	Val	0.21232877	0.20319635	0.21689498	0.1826484	0.16666667	0.15068493	0.1369863	0.12328767	0.07762557	0.08447489	0.10045662	0.0913242	0.06164384
			Test	0.74180328	0.62704918	0.67213115	0.64344262	0.67622951	0.68852459	0.68852459	0.65163934	0.72540984	0.68442623	0.67622951	0.65983607	0.39754098
		Acc0_Dat1	Val	0.02970297	0.01980198	0.0049505	0.01485149	0.0049505	0.0049505	0.0049505	0.0049505	0.0049505	0.0049505	0	0	0
			Test	0.015625	0.03125	0.01041667	0.01041667	0.01041667	0.00520833	0.00520833	0	0	0	0	0	0
		Nom0_Dat1	Val	0.01485149	0.01485149	0.00990099	0.00990099	0	0	0	0	0	0	0	0	0
			Test	0.06770833	0.08854167	0.08333333	0.06770833	0.04166667	0.03645833	0.03645833	0.02083333	0	0	0	0	0.00520833
	Cased	Nom1_Acc0_Dat1	Val	0.16831683	0.15511551	0.1320132	0.11881188	0.08250825	0.0660066	0.05610561	0.0330033	0.01320132	0.01650165	0.01320132	0.00660066	0.00990099
			Test	0.03645833	0.046875	0.03645833	0.02604167	0.01041667	0	0	0	0	0	0	0	0
		Nom0_Acc0_Dat1	Val	0.02970297	0.03630363	0.01650165	0.01320132	0.00990099	0.00990099	0.00660066	0.00660066	0.00660066	0.00660066	0.0033003	0.0033003	0
			Test	0.04166667	0.046875	0.03645833	0.02083333	0.03125	0.015625	0.02604167	0.015625	0	0	0	0	0
		Nom0_Acc1_Dat1	Val	0.13861386	0.11881188	0.0990099	0.07260726	0.04620462	0.0330033	0.02970297	0.01980198	0.00990099	0.00660066	0.00660066	0.00660066	0.00660066
Noun Only			Test	0.1875	0.16666667	0.18229167	0.078125	0.05729167	0.0625	0.05208333	0.015625	0.00520833	0	0	0.00520833	0.015625
		Acc0_Dat1	Val	0.16831683	0.13366337	0.13366337	0.11386139	0.1039604	0.11386139	0.06435644	0.06930693	0.02970297	0.05445545	0.02970297	0.01485149	0.02970297
			Test	0.109375	0.17708333	0.21875	0.24479167	0.21875	0.17708333	0.22395833	0.1875	0.203125	0.21875	0.1875	0.02604167	0.03125
		Nom0_Dat1	Val	0.19306931	0.18316832	0.17326733	0.15841584	0.12871287	0.11881188	0.08910891	0.03960396	0.01485149	0.02475248	0.03960396	0.03465347	0.03465347
			Test	0.69270833	0.578125	0.609375	0.609375	0.67708333	0.57291667	0.609375	0.63020833	0.55208333	0.67708333	0.68229167	0.375	0.33854167
	Caseless	Nom1_Acc0_Dat1	Val	0.22442244	0.21122112	0.17821782	0.15181518	0.13861386	0.13861386	0.11221122	0.09570957	0.08250825	0.07920792	0.06270627	0.0330033	0.04950495
			Test	0.05729167	0.0625	0.05729167	0.04166667	0.046875	0.03125	0.04166667	0.01041667	0	0	0.00520833	0	0
		Nom0_Acc0_Dat1	Val	0.19471947	0.18151815	0.18481848	0.15181518	0.11551155	0.1320132	0.07920792	0.05280528	0.04620462	0.04290429	0.02970297	0.02970297	0.0330033
			Test	0.515625	0.421875	0.41666667	0.48958333	0.67708333	0.44791667	0.5625	0.47916667	0.390625	0.56770833	0.55729167	0.25	0.234375
		Nom0_Acc1_Dat1	Val	0.26732673	0.22772277	0.20792079	0.18151815	0.15841584	0.15181518	0.11881188	0.05940594	0.04620462	0.04950495	0.04290429	0.02310231	0.03630363
			Test	0.71354167	0.72395833	0.71354167	0.77083333	0.8125	0.80729167	0.79166667	0.82291667	0.81770833	0.73958333	0.83854167	0.671875	0.41666667
		Acc0_Dat1	Val	0	0	0	0	0	0	0	0	0	0	0	0	0
			Test	0	0	0	0	0	0	0	0	0	0	0	0	0.01923076923
		Nom0_Dat1	Val	0	0	0	0	0	0	0	0	0	0	0	0	0
			Test	0	0	0	0	0	0.01923076923	0.01923076923	0.03846153846	0.03846153846	0.01923076923	0	0	0.01923076923
	Cased	Nom1_Acc0_Dat1	Val	0.1666666667	0.1166666667	0.08333333	0.05	0.06666667	0.03333333	0.05	0.03333333	0.01666666667	0.01666666667	0	0	0.01666666667
			Test	0	0	0	0	0	0	0	0	0	0	0	0	0
		Nom0_Acc0_Dat1	Val	0	0	0	0	0	0	0	0	0	0	0	0	0
			Test	0	0	0	0	0.01923076923	0.01923076923	0	0.01923076923	0	0.01923076923	0	0	0.01923076923
		Nom0_Acc1_Dat1	Val	0.2	0.2	0.1	0.08333333	0.08333333	0.1	0.06666667	0.01666666667	0	0	0	0	0
Pronoun Only			Test	0.01923076923	0.03846153846	0.05769230769	0.07692308	0.07692308	0.01923076923	0.05769230769	0.01923076923	0.01923076923	0.01923076923	0.01923076923	0	0.01923076923
		Acc0_Dat1	Val	0.075	0.1	0.075	0.125	0.1	0.15	0.075	0.125	0.075	0.125	0.1	0.15	0.1
			Test	0.23076923	0.26923077	0.25	0.23076923	0.23076923	0.25	0.17307692	0.15384615	0.09615385	0.28846154	0.36538462	0.17307692	0.23076923
		Nom0_Dat1	Val	0.2	0.075	0.075	0.05	0.05	0.05	0.05	0.075	0.05	0.1	0.075	0.1	0.05
			Test	0.40384615	0.40384615	0.48076923	0.40384615	0.36538462	0.42307692	0.36538462	0.42307692	0.46153846	0.48076923	0.55769231	0.44230769	0.28846154
	Caseless	Nom1_Acc0_Dat1	Val	0.25	0.3	0.2	0.26666667	0.26666667	0.21666667	0.26666667	0.28333333	0.16666667	0.16666667	0.2	0.16666667	0.23333333
			Test	0.19230769	0.21153846	0.25	0.23076923	0.21153846	0.19230769	0.15384615	0.15384615	0.07692308	0.09615385	0.11538462	0.09615385	0.19230769
		Nom0_Acc0_Dat1	Val	0.15	0.08333333	0.11666667	0.05	0.08333333	0.1	0.06666667	0.06666667	0.03333333	0.06666667	0.08333333	0.05	0.03333333
			Test	0.34615385	0.36538462	0.32692308	0.40384615	0.36538462	0.32692308	0.34615385	0.25	0.34615385	0.42307692	0.59615385	0.19230769	0.34615385
		Nom0_Acc1_Dat1	Val	0.4	0.2833333	0.25	0.23333333	0.25	0.15	0.2	0.25	0.21666667	0.2	0.21666667	0.2	0.16666667
			Test	0.61538462	0.51923077	0.51923077	0.44230769	0.40384615	0.48076923	0.48076923	0.51923077	0.48076923	0.48076923	0.48076923	0.61538462	0.42307692
		Acc_random	Val	-												
			Test	0.47131148	0.6147541	0.44262295	0.57377049	0.50819672	0.41393443	0.4795082	0.38934426	0.31557377	0.56557377	0.38934426	0.60245902	0.46721311
	Cased	Nom random	Val	-												
			Test	0.55737705	0.5204918	0.62704918	0.56147541	0.52459016	0.40983607	0.55327869	0.54918033	0.50819672	0.45081967	0.74590164	0.44672131	0.45491803
		All Random	Val	-												
Control			Test	0.60655738	0.60655738	0.42213115	0.54508197	0.3852459	0.48770492	0.47131148	0.55327869	0.49180328	0.44262295	0.68032787	0.26639344	0.41393443
		Acc_random	Val	-												
			Test	0.38114754	0.61885246	0.67213115	0.63934426	0.39754098	0.38934426	0.42213115	0.4795082	0.40983607	0.17213115	0.17213115	0.26639344	0.35655738
	Caseless	Nom random	Val	-												
			Test	0.44672131	0.50819672	0.45901639	0.45901639	0.52868852	0.52459016	0.52868852	0.53278689	0.55737705	0.45491803	0.52459016	0.35655738	0.5
		All Random	Val	-												
			Test	0.43032787	0.46311475	0.57377049	0.48770492	0.44262295	0.41803279	0.54918033	0.54098361	0.54508197	0.38114754	0.45081967	0.28688525	0.45901639

Table 2: Full results for Bert model. Including the layer by layer test error rate, the layer by layer test error rates for the control datasets (Same Token Amount, Noun Only, and Pronoun Only), and the layer by layer test error rate for the control probe. The table also includes all error rates for the validation set, minus the control probe where there was no validation set. The results reported in the main paper are from the Random dataset, layer 11.

B Full mBert Results

				layers												
				0	1	2	3	4	5	6	7	8	9	10	11	12
		Acc0_Dat1	Val	0.015	0.0125	0.0125	0.01	0.01	0.0125	0.01	0.0075	0.01	0.005	0.0025	0.005	0.01
			Test	0.01229508	0.01229508	0.01639344	0.01639344	0.00819672	0.00409836	0.00409836	0.00409836	0.01639344	0.01229508	0.02459016	0.02459016	0.02868852
		Nom0_Dat1	Val	0.0275	0.02	0.02	0.0125	0.01	0.005	0.0075	0.005	0.005	0.005	0.005	0.0075	0
			Test	0.09836066	0.10245902	0.06147541	0.04918033	0.04098361	0.03278689	0.01229508	0.01229508	0.03688525	0.04098361	0.08196721	0.05327869	0.08196721
	Cased	Nom1_Acc0_Dat1	Val	0.10833333	0.11833333	0.105	0.09166667	0.08833333	0.05166667	0.04333333	0.03166667	0.01166667	0.02	0.02166667	0.02333333	0.03833333
			Test	0.04098361	0.02459016	0.02459016	0.02459016	0.01639344	0.01229508	0	0	0	0	0.00409836	0.00409836	0.00819672
		Nom0_Acc0_Dat1	Val	0.03666667	0.03	0.02333333	0.01833333	0.01	0.01166667	0.01	0.00833333	0.00666667	0.005	0.00333333	0.005	0.00833333
			Test	0.10655738	0.08196721	0.04508197	0.04098361	0.04918033	0.0204918	0.03688525	0.01639344	0.02868852	0.05327869	0.05327869	0.05737705	0.06967213
		Nom0_Acc1_Dat1	Val	0.11666667	0.10333333	0.12	0.10666667	0.09666667	0.06833333	0.03166667	0.02	0.02	0.02666667	0.025	0.03333333	0.03666667
			Test	0.18032787	0.16803279	0.15983607	0.12704918	0.09836066	0.11065574	0.10655738	0.05737705	0.08196721	0.08606557	0.11885246	0.12704918	0.17213115
Random		Acc0_Dat1	Val	0.1775	0.155	0.1425	0.135	0.125	0.1225	0.0975	0.085	0.0875	0.08	0.08	0.0925	0.09
			Test	0.21311475	0.14344262	0.17622951	0.16803279	0.15983607	0.17213115	0.2295082	0.20901639	0.34836066	0.30737705	0.30737705	0.29098361	0.19262295
		Nom0_Dat1	Val	0.27	0.2325	0.23	0.205	0.17	0.1725	0.13	0.12	0.0975	0.085	0.0975	0.1075	0.105
			Test	0.61065574	0.56147541	0.53278689	0.57786885	0.57377049	0.55327869	0.62704918	0.72540984	0.63934426	0.62704918	0.62295082	0.50409836	0.56147541
	Caseless	Nom1_Acc0_Dat1	Val	0.22	0.20666667	0.18833333	0.175	0.17666667	0.17166667	0.15333333	0.12	0.14	0.11833333	0.13333333	0.13166667	0.13333333
			Test	0.12295082	0.06967213	0.06557377	0.06557377	0.07377049	0.06967213	0.06967213	0.01639344	0.03278689	0.02868852	0.03688525	0.03688525	0.04508197
		Nom0_Acc0_Dat1	Val	0.26333333	0.21666667	0.20166667	0.19	0.16666667	0.14833333	0.12833333	0.11833333	0.115	0.11666667	0.09666667	0.11	0.12
			Test	0.48770492	0.37704918	0.42213115	0.35245902	0.45491803	0.39754098	0.39754098	0.68032787	0.64344262	0.56147541	0.60655738	0.53688525	0.53688525
		Nom0_Acc1_Dat1	Val	0.23	0.24	0.23	0.2266667	0.215	0.2	0.16	0.14666667	0.12833333	0.12333333	0.11333333	0.15	0.14166667
			Test	0.68442623	0.68032787	0.71311475	0.70491803	0.68442623	0.65983607	0.68032787	0.76639344	0.78278689	0.62704918	0.68442623	0.69262295	0.68442623
		Acc0_Dat1	Val	0.03767123	0.03767123	0.03424658	0.02739726	0.01712329	0.01369863	0.01712329	0.01369863	0.01027397	0.01369863	0.01027397	0.01027397	0.01712329
			Test	0.03278689	0.01639344	0.02459016	0.0204918	0.00819672	0.00409836	0	0.00409836	0.01639344	0.0204918	0.02868852	0.03688525	0.02459016
		Nom0_Dat1	Val	0.02054795	0.02054795	0.01712329	0.01027397	0.00684932	0.00342466	0.00342466	0.00342466	0	0.00342466	0	0	0.01027397
			Test	0.1147541	0.14344262	0.1352459	0.10655738	0.08606557	0.04508197	0.03688525	0.04098361	0.08196721	0.06557377	0.10245902	0.10245902	0.10245902
	Cased	Nom1_Acc0_Dat1	Val	0.12328767	0.11643836	0.11187215	0.10502283	0.07762557	0.05251142	0.02283105	0.02511416	0.02739726	0.01826484	0.02739726	0.03196347	0.043379
			Test	0.04508197	0.0204918	0.0204918	0.03278689	0.01639344	0.00819672	0	0	0	0	0	0.00409836	0.00409836
		Nom0_Acc0_Dat1	Val	0.04794521	0.0456621	0.03881279	0.03196347	0.01826484	0.02283105	0.00684932	0.00456621	0.00913242	0.00456621	0.00684932	0.00684932	0.01369863
			Test	0.06557377	0.07377049	0.06557377	0.06967213	0.04918033	0.03688525	0.04098361	0.03688525	0.04098361	0.05327869	0.06147541	0.10245902	0.0942623
		Nom0_Acc1_Dat1	Val	0.15068493	0.11415525	0.10502283	0.09360731	0.06849315	0.05251142	0.01369863	0.01598174	0.01141553	0.01826484	0.02054795	0.02511416	0.03196347
			Test	0.24590164	0.28688525	0.26639344	0.17213115	0.14344262	0.1352459	0.11885246	0.09836066	0.15983607	0.08606557	0.1147541	0.14754098	0.22131148
Same Token Amount		Acc0_Dat1	Val	0.21575342	0.17123288	0.15753425	0.17465753	0.15410959	0.16438356	0.14383562	0.14041096	0.1130137	0.09589041	0.11986301	0.1130137	0.11986301
			Test	0.23360656	0.12704918	0.17622951	0.12704918	0.19672131	0.13114754	0.12295082	0.14754098	0.21721311	0.30737705	0.25409836	0.27459016	0.19262295
		Nom0_Dat1	Val	0.23287671	0.22945205	0.19863014	0.16780822	0.1369863	0.1369863	0.12671233	0.10273973	0.0890411	0.06849315	0.10273973	0.11643836	0.10616438
			Test	0.68032787	0.57786885	0.57377049	0.58196721	0.61885246	0.55737705	0.58196721	0.71311475	0.7295082	0.61885246	0.55737705	0.54098361	0.55737705
	Caseless	Nom1_Acc0_Dat1	Val	0.2283105	0.23059361	0.21004566	0.19634703	0.17351598	0.16894977	0.15068493	0.1347032	0.12100457	0.10502283	0.12328767	0.14383562	0.15296804
			Test	0.1147541	0.06147541	0.06147541	0.07377049	0.06967213	0.04918033	0.06147541	0.01229508	0.02868852	0.04918033	0.03278689	0.02459016	0.02868852
		Nom0_Acc0_Dat1	Val	0.2739726	0.23744292	0.22146119	0.20776256	0.19634703	0.18493151	0.14611872	0.12557078	0.13013699	0.1347032	0.1369863	0.1347032	0.14383562
			Test	0.52459016	0.45491803	0.43852459	0.41803279	0.48360656	0.44672131	0.38114754	0.60245902	0.66393443	0.54508197	0.61065574	0.54098361	0.52868852
		Nom0_Acc1_Dat1	Val	0.21689498	0.21917808	0.2260274	0.18721461	0.19178082	0.15981735	0.1369863	0.10273973	0.11187215	0.11187215	0.11415525	0.14383562	0.1347032
			Test	0.78278689	0.67213115	0.70901639	0.74590164	0.7295082	0.72540984	0.78278689	0.78278689	0.82786885	0.75819672	0.70901639	0.68442623	0.67213115
		Acc0_Dat1	Val	0.03465347	0.04950495	0.03465347	0.01980198	0.01980198	0.00990099	0.00990099	0.00990099	0.00990099	0.0049505	0.01485149	0.01485149	0.01980198
			Test	0.03645833	0.02604167	0.03125	0.02604167	0.01041667	0.00520833	0.015625	0.01041667	0.015625	0.02083333	0.03125	0.05208333	0.03645833
		Nom0_Dat1	Val	0.03465347	0.02475248	0.02475248	0.01980198	0.01485149	0.0049505	0	0	0.0049505	0	0	0	0
			Test	0.11979167	0.11458333	0.09895833	0.08333333	0.07291667	0.07291667	0.05729167	0.078125	0.14583333	0.11979167	0.140625	0.20833333	0.171875
	Cased	Nom1_Acc0_Dat1	Val	0.16171617	0.1650165	0.15841584	0.14521452	0.12211221	0.09240924	0.04620462	0.01650165	0.00990099	0.01320132	0.02640264	0.0330033	0.04290429
			Test	0.04166667	0.015625	0.02083333	0.03125	0.015625	0	0.01041667	0	0	0	0	0	0
		Nom0_Acc0_Dat1	Val	0.02970297	0.03960396	0.03630363	0.02970297	0.01650165	0.00990099	0.00660066	0.00660066	0.00660066	0	0.00990099	0.01320132	0.02640264
			Test	0.078125	0.05729167	0.046875	0.05208333	0.046875	0.03125	0.05729167	0.03645833	0.13020833	0.09895833	0.10416667	0.15104167	0.13020833
		Nom0_Acc1_Dat1	Val	0.21452145	0.15511551	0.15181518	0.12541254	0.10231023	0.06270627	0.02640264	0.02970297	0.02310231	0.01980198	0.01980198	0.01980198	0.04290429
			Test	0.22916667	0.27083333	0.27604167	0.203125	0.140625	0.18229167	0.17708333	0.10416667	0.10416667	0.078125	0.13541667	0.17708333	0.23958333
Noun Only		Acc0_Dat1	Val	0.1980198	0.13861386	0.13861386	0.13366337	0.13366337	0.11386139	0.11386139	0.07920792	0.06435644	0.06930693	0.05445545	0.05940594	0.07920792
			Test	0.22916667	0.13541667	0.125	0.16666667	0.11979167	0.14583333	0.15104167	0.14583333	0.19791667	0.24479167	0.23958333	0.24479167	0.15104167
		Nom0_Dat1	Val	0.23762376	0.19306931	0.2029703	0.17326733	0.15346535	0.15841584	0.09405941	0.07425743	0.05940594	0.05445545	0.04950495	0.04455446	0.06930693
			Test	0.63541667	0.61458333	0.6875	0.640625	0.6875	0.64583333	0.67708333	0.80729167	0.703125	0.6875	0.66145833	0.66666667	0.64583333
	Caseless	Nom1_Acc0_Dat1	Val	0.22112211	0.16831683	0.15841584	0.17821782	0.17491749	0.14191419	0.13531353	0.07590759	0.08250825	0.08910891	0.09570957	0.10231023	0.10231023
			Test	0.06770833	0.046875	0.046875	0.046875	0.04166667	0.04166667	0.02604167	0.00520833	0	0.00520833	0.015625	0.015625	0.015625
		Nom0_Acc0_Dat1	Val	0.22442244	0.20132013	0.19471947	0.19141914	0.15841584	0.17161716	0.15181518	0.08250825	0.09570957	0.08580858	0.08910891	0.08250825	0.09240924
			Test	0.53125	0.48958333	0.5	0.4375	0.52604167	0.46354167	0.44270833	0.67708333	0.52604167	0.609375	0.57291667	0.57291667	0.51041667
		Nom0_Acc1_Dat1	Val	0.28052805	0.23762376	0.21782178	0.18811881	0.17491749	0.14191419	0.0990099	0.06930693	0.04950495	0.08580858	0.08580858	0.10891089	0.10891089
			Test	0.765625	0.796875	0.81770833	0.77083333	0.83333333	0.83854167	0.875	0.89583333	0.859375	0.73958333	0.765625	0.76041667	0.73958333
		Acc0_Dat1	Val	0.025	0.025	0.05	0.05	0.075	0.05	0.075	0.05	0.05	0.05	0.025	0.025	0
			Test	0	0	0	0.01923077	0.01923077	0.01923077	0	0	0.01923077	0	0.01923077	0	0
		Nom0_Dat1	Val	0	0	0	0	0	0	0	0	0	0	0	0	0
			Test	0	0	0	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.03846154	0.03846154	0.03846154	0.05769231
	Cased	Nom1_Acc0_Dat1	Val	0.18333333	0.21666667	0.18333333	0.18333333	0.16666667	0.13333333	0.08333333	0.05	0.06666667	0.03333333	0.06666667	0.05	0.08333333
			Test	0	0.03846154	0.01923077	0.01923077	0.01923077	0.01923077	0	0	0	0	0	0	0.01923077
		Nom0_Acc0_Dat1	Val	0	0	0.01666667	0	0	0	0.01666667	0	0	0	0	0	0
			Test	0	0	0	0.01923077	0	0	0	0	0.01923077	0.03846154	0.07692308	0.07692308	0
		Nom0_Acc1_Dat1	Val	0.16666667	0.16666667	0.13333333	0.16666667	0.05	0.05	0.05	0.03333333	0.01666667	0.01666667	0.01666667	0	0.01666667
			Test	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.03846154	0.03846154	0.03846154	0.03846154	0.03846154	0.01923077	0.01923077	0.09615385
Pronoun Only		Acc0_Dat1	Val	0.1	0.125	0.1	0.125	0.175	0.15	0.125	0.15	0.15	0.175	0.125	0.15	0.125
			Test	0.23076923	0.17307692	0.17307692	0.17307692	0.19230769	0.21153846	0.21153846	0.23076923	0.30769231	0.36538462	0.40384615	0.34615385	0.44230769
		Nom0_Dat1	Val	0.2	0.05	0.1	0.075	0.125	0.1	0.125	0.15	0.175	0.125	0.15	0.075	0.075
			Test	0.40384615	0.26923077	0.30769231	0.26923077	0.21153846	0.32692308	0.38461538	0.51923077	0.53846154	0.59615385	0.57692308	0.53846154	0.55769231
	Caseless	Nom1_Acc0_Dat1	Val	0.21666667	0.18333333	0.21666667	0.18333333	0.26666667	0.25	0.21666667	0.2	0.2	0.16666667	0.2	0.21666667	0.2
			Test	0.26923077	0.19230769	0.19230769	0.25	0.13461538	0.25	0.21153846	0.23076923	0.19230769	0.21153846	0.21153846	0.21153846	0.21153846
		Nom0_Acc0_Dat1	Val	0.18333333	0.13333333	0.13333333	0.15	0.16666667	0.18333333	0.13333333	0.11666667	0.16666667	0.13333333	0.16666667	0.11666667	0.11666667
			Test	0.42307692	0.26923077	0.21153846	0.26923077	0.25	0.23076923	0.30769231	0.40384615	0.5	0.42307692	0.57692308	0.5	0.53846154
		Nom0_Acc1_Dat1	Val	0.33333333	0.21666667	0.3	0.3	0.31666667	0.31666667	0.36666667	0.23333333	0.18333333	0.26666667	0.25	0.21666667	0.3
			Test	0.48076923	0.36538462	0.46153846	0.44230769	0.53846154	0.5	0.53846154	0.65384615	0.55769231	0.61538462	0.57692308	0.67307692	0.59615385
		Acc_random	Val	-												
			Test	0.5942623	0.48360656	0.45491803	0.42213115	0.54918033	0.44672131	0.46721311	0.34016393	0.62704918	0.41803279	0.38934426	0.30737705	0.43852459
	Cased	Nom random	Val	-												
			Test	0.40163934	0.42622951	0.52459016	0.47131148	0.44262295	0.45491803	0.61065574	0.5204918	0.35655738	0.37295082	0.43442623	0.37704918	0.33606557
		All Random	Val	-												
Control			Test	0.51229508	0.49180328	0.38114754	0.4057377	0.43032787	0.37704918	0.52459016	0.36885246	0.46311475	0.39344262	0.44262295	0.40983607	0.3647541
		Acc_random	Val	-												
			Test	0.53688525	0.54508197	0.44672131	0.5204918	0.49180328	0.49590164	0.40163934	0.14344262	0.54508197	0.40163934	0.2295082	0.29508197	0.22540984
	Caseless	Nom random	Val	-												
			Test	0.47131148	0.48360656	0.41393443	0.48770492	0.3852459	0.27459016	0.23360656	0.43852459	0.4795082	0.5	0.52459016	0.54098361	0.3442623
		All Random	Val	-												
			Test	0.45081967	0.39754098	0.3647541	0.56147541	0.33606557	0.26229508	0.26229508	0.30737705	0.34836066	0.20491803	0.3647541	0.24590164	0.30327869

Table 3: Full results for mBert model. Including the layer by layer test error rate, the layer by layer test error rates for the control datasets (Same Token Amount, Noun Only, and Pronoun Only), and the layer by layer test error rate for the control probe. The table also includes all error rates for the validation set, minus the control probe where there was no validation set. The results reported in the main paper are from the Random dataset, layer 10.

C Full LläMmlein 1B Results

				layers																							
				0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Random	Acc0_Dat1		Val	0.0275	0.0225	0.015	0.03	0.0225	0.0075	0.015	0.01	0.01	0.0075	0.01	0.0225	0.01	0.0075	0.0025	0.0025	0.0025	0.0025	0.0025	0.0025	0	0.005	0.0025	0
			Test	0.04098361	0.06557377	0.03278689	0.0204918	0.04508197	0.01229508	0.045082197	0	0.00409836	0.00409836	0.00409836	0.00409836	0.01639344	0.00819672	0.00819672	0.00819672	0.00819672	0.00819672	0.00819672	0.00819672	0.00819672	0.00819672	0.00819672	0
	Acc0_Dat1	Cased	Val	0.0525	0.035	0.05	0.03	0.0175	0.015	0.015	0.005	0.005	0.005	0.0025	0.0025	0.0025	0.0025	0.005	0	0	0	0	0	0	0	0.0025	0.0075	0.0025
			Test	0.13114754	0.15983607	0.19262295	0.04508197	0.05737705	0.07377049	0.03278689	0.02459016	0.02049918	0.03278689	0.02459016	0.04508197	0.03278689	0.02868852	0.0204918	0.0204918	0.0204918	0.02049180	0.01639344	0.0204918	0.01639344	0.0204918	0.01229508	0.01639344	0
	Non0_Dat0_Dat1		Val	0.15	0.13666667	0.13666667	0.11166667	0.08833333	0.09166667	0.03833333	0.03166667	0.045	0.035	0.03	0.03333333	0.03333333	0.04166667	0.02166667	0.02333333	0.025	0.02333333	0.02333333	0.02	0.03666667	0.02333333	0.04	
			Test	0.04508197	0.02868852	0.0204918	0.0204918	0.0204918	0.01639344	0.04098361	0.03278689	0.00819672	0.01639344	0.02868852	0.02459016	0.03688525	0.03688525	0.00409836	0	0	0	0.00409836	0.00409836	0	0.00409836	0.00409836		
	Non0_Dat0_Dat1		Val	0.03666667	0.05166667	0.01666667	0.03	0.03	0.03	0.01666667	0.01666667	0.005	0.00666667	0.00666667	0.00666667	0.00333333	0.005	0.005	0.00333333	0.00166667	0.00166667	0.00333333	0.00166667	0.00166667	0.00333333	0.00166667	0.01	
			Test	0.17622951	0.15983607	0.1352459	0.06147541	0.07377049	0.06967213	0.06967213	0.02459016	0.0204918	0.02459016	0.01229508	0.03278689	0.02868852	0.00819672	0.01639344	0.01639344	0.01639344	0.01639344	0.01639344	0.01639344	0.01639344	0.01639344	0.01229508	0.0204918	
	Non0_Acc1_Dat1		Val	0.15666667	0.125	0.095	0.095	0.07166667	0.06666667	0.025	0.02166667	0.02166667	0.01666667	0.01666667	0.02833333	0.025	0.03	0.0183333	0.02166667	0.025	0.0183333	0.02166667	0.02	0.02833333	0.02833333	0.02833333		
			Test	0.19672131	0.19262295	0.18032787	0.18442623	0.16803279	0.16393443	0.10655738	0.06147541	0.02459016	0.02459016	0.03278689	0.06967213	0.06147541	0.04098361	0.01639344	0.01229508	0.0204918	0.02049180	0.02868852	0.0204918	0.01639344	0.0204918	0.02049180	0.0204918	
	Acc0_Dat1		Val	0.27	0.34	0.225	0.21	0.2275	0.185	0.175	0.18	0.1525	0.145	0.135	0.14	0.1325	0.1275	0.115	0.11	0.105	0.105	0.1025	0.075	0.08	0.0775	0.095	
			Test	0.07786885	0.4057377	0.08606557	0.09836066	0.08606557	0.07786885	0.08196721	0.06967213	0.05327869	0.05327869	0.06967213	0.07786885	0.07786885	0.07377049	0.07868852	0.2295082	0.18442623	0.29918033	0.28278689	0.1352459	0.1352459	0.12295082	0.08196721	
	Non0_Dat1		Val	0.2875	0.2625	0.2625	0.26	0.2325	0.2475	0.1575	0.27	0.2025	0.205	0.175	0.155	0.15	0.1475	0.1375	0.15	0.1625	0.155	0.1075	0.135	0.1425	0.12	0.115	
			Test	0.6557377	0.75409836	0.82377049	0.74180328	0.7704918	0.82377049	0.72540984	0.8647541	0.68852459	0.71721311	0.70081967	0.69262295	0.68852459	0.70491803	0.70491803	0.72131148	0.7295082	0.72540984	0.76639344	0.68852459	0.66803279	0.67213115	0.71311475		
	Non1_Acc0_Dat1	Caseless	Val	0.285	0.235	0.23666667	0.26	0.25	0.22166667	0.22333333	0.21333333	0.21166667	0.22333333	0.2283333	0.165	0.17333333	0.205	0.18	0.16	0.16166667	0.13666667	0.14833333	0.1833333	0.13166667	0.12833333	0.1183333	
			Test	0.08196721	0.06557377	0.08196721	0.11065574	0.10245902	0.09016393	0.07786885	0.05737705	0.06967213	0.04508197	0.06147541	0.05737705	0.08196721	0.04508197	0.04918033	0.03688525	0.06557377	0.04508197	0.05737705	0.02868852	0.04098361	0.04098361	0.02459016		
	Non0_Dat0_Dat1		Val	0.38833333	0.41166667	0.41166667	0.42166667	0.41166667	0.38833333	0.2	0.23166667	0.31166667	0.205	0.19333333	0.18333333	0.195	0.18333333	0.17833333	0.17666667	0.16166667	0.13333333	0.165	0.14666667	0.14166667	0.11	0.12166667	
			Test	0.54098361	0.51639344	0.58196721	0.54508197	0.52868852	0.50409836	0.70901639	0.76639344	0.58196721	0.70081967	0.56967213	0.58606557	0.6557377	0.70491803	0.69672131	0.68442623	0.54508197	0.60245902	0.69334426	0.56147541	0.66803279	0.56147541	0.55327869		
	Non0_Acc1_Dat1		Val	0.25333333	0.24	0.23666667	0.28333333	0.26833333	0.23833333	0.22166667	0.19833333	0.215	0.21166667	0.19166667	0.17	0.16833333	0.15666667	0.14	0.13833333	0.15333333	0.125	0.13666667	0.12	0.13166667	0.13333333	0.12		
			Test	0.79098361	0.81557377	0.85655738	0.86065574	0.85245902	0.8442623	0.85655738	0.82786885	0.81147541	0.84016393	0.78688525	0.75819672	0.76229508	0.78688525	0.76639344	0.80737705	0.78278689	0.7704918	0.78278689	0.85655738	0.77868852	0.75409836	0.79918033		
	Acc0_Dat1		Val	0.06164384	0.07191781	0.05821918	0.06849315	0.04452055	0.03424658	0.0239726	0.00684932	0.00684932	0.01369863	0.01369863	0.00684932	0.01369863	0.01027397	0.00342466	0.00342466	0.00342466	0.00342466	0.00684932	0.00342466	0.00684932	0.01027397	0.01369863	
			Test	0.04508197	0.04918033	0.03688525	0.0204918	0.03688525	0.03278689	0.04098361	0.01229508	0.00819672	0.00819672	0.01639344	0.01639344	0.01639344	0.00819672	0.00819672	0.00819672	0.00819672	0.00819672	0.00819672	0.00819672	0.01229508	0.01639344	0.01639344		
	Non0_Dat1		Val	0.06849315	0.07191781	0.04109589	0.04452055	0.02054795	0.00684932	0.01027397	0	0.00684932	0	0	0	0	0	0	0	0.00342466	0.00342466	0.00342466	0.00342466	0.00342466	0	0	0
			Test	0.2295082	0.19672131	0.17213115	0.19672131	0.08196721	0.04918033	0.07377049	0.03688525	0.03278689	0.04098361	0.02459016	0.02868852	0.03278689	0.0204918	0.01229508	0.01639344	0.01639344	0.01229508	0.01229508	0.01229508	0.01639344	0.01639344	0.01639344		
	Non1_Acc0_Dat1		Val	0.11872146	0.13470032	0.10958904	0.11415525	0.0890411	0.08219178	0.06164384	0.05251142	0.06164384	0.04109589	0.05251142	0.03881279	0.03652968	0.05251142	0.03881279	0.0319635	0.03652968	0.02739726	0.03652968	0.02739726	0.04109589	0.05251142	0.05707763		
		Cased	Test	0.03527869	0.04098361	0.01639344	0.0204918	0.03278689	0.00819672	0.02868852	0.0204918	0.00409836	0.00409836	0.00409836	0.00819672	0.00819672	0.02459016	0.00819672	0.00819672	0.00819672	0.01229508	0.00819672	0.00819672	0.00819672	0.01229508	0.0204918		
	Non0_Dat0_Dat1		Val	0.05936073	0.07990868	0.06164384	0.07762557	0.05936073	0.01598174	0.02511416	0.00456621	0.01141553	0.00456621	0.00684932	0.00456621	0.00913242	0.00913242	0.00456621	0.00228311	0.00228311	0.00456621	0.00456621	0.00456621	0.00913242	0.01141553		
			Test	0.13934426	0.13032787	0.16803279	0.10245902	0.06967213	0.04918033	0.05327869	0.02868852	0.00819672	0.01639344	0.03688525	0.03278689	0.0204918	0.01639344	0.01639344	0.01639344	0.0204918	0.01229508	0.01229508	0.01639344	0.01639344	0.01639344	0.01639344		
	Non0_Acc1_Dat1		Val	0.14383562	0.14383562	0.11872146	0.0890411	0.05479452	0.02511142	0.04794521	0.02511416	0.03424658	0.03196347	0.03196347	0.03424658	0.03196347	0.02739726	0.02511416	0.02968037	0.03196347	0.02968037	0.02739726	0.02511416	0.03424658	0.02511416	0.04109589		
			Test	0.19262295	0.18442623	0.18442623	0.22540984	0.20901639	0.20491803	0.18032787	0.10655738	0.08606557	0.08606557	0.07377049	0.08606557	0.07786885	0.08606557	0.07786885	0.08606557	0.01639344	0.01639344	0.01639344	0.01229508	0.01229508	0.01229508	0.0204918	0.02459016	0.03688525	
Same Token Amount			Val	0.30479452	0.35273973	0.32191781	0.30136986	0.29109589	0.27054795	0.26712329	0.2260274	0.23972603	0.23630137	0.21575342	0.20547945	0.21232877	0.21917808	0.18150685	0.17123288	0.14383562	0.18150685	0.14383562	0.14041096	0.13013699	0.12671233	0.13356164		
	Acc0_Dat1		Test	0.07786885	0.45081967	0.09016393	0.07786885	0.07377049	0.07786885	0.0942623	0.07786885	0.07377049	0.07786885	0.07377049	0.08606557	0.09836066	0.08606557	0.31557377	0.08196721	0.34016393	0.10655738	0.33606557	0.1557377	0.23770492	0.20081967	0.07737049		
			Val	0.29452055	0.26712329	0.25	0.24657534	0.23972603	0.23287671	0.2260274	0.25	0.26027397	0.19863014	0.19863014	0.17123288	0.15753425	0.17465753	0.1609589	0.14726027	0.1369863	0.13356164	0.1369863	0.13013699	0.14383562	0.14041096	0.13356164		
	Non0_Dat1		Test	0.70901639	0.79098361	0.81147541	0.84967213	0.76229508	0.82786885	0.82786885	0.86065574	0.75	0.79081967	0.68442623	0.68852459	0.71721311	0.70081967	0.74180328	0.73360656	0.72540984	0.7295082	0.69672131	0.73360656	0.73770492	0.71311475	0.73770492		
			Val	0.27625571	0.29908676	0.30136986	0.27853881	0.29223744	0.26711329	0.25342466	0.26255708	0.26484018	0.25799087	0.23744292	0.23744292	0.23515982	0.23972603	0.20776256	0.20547945	0.19406393	0.19634703	0.20319635	0.13926941	0.1803653	0.16666667	0.1347032		
	Non1_Acc0_Dat1	Caseless	Test	0.08606557	0.08196721	0.07377049	0.09016393	0.10655738	0.0942623	0.07377049	0.06557377	0.06967213	0.05737705	0.06557377	0.07786885	0.06967213	0.06967213	0.06147541	0.05737705	0.06147541	0.06557377	0.06147541	0.0204918	0.05327869	0.05737705	0.0204918		
			Val	0.42922374	0.42922374	0.37671233	0.44173516	0.42694064	0.42465753	0.41552511	0.21917808	0.31050228	0.2146187	0.32191781	0.20091324	0.22146119	0.18721461	0.23287671	0.18721461	0.1803653	0.13926941	0.17123288	0.16894977	0.16666667	0.17808219	0.13013699		
	Non0_Acc1_Dat1		Test	0.54508197	0.54508197	0.60655738	0.49590164	0.6147541	0.59836066	0.65163934	0.75	0.61065574	0.7704918	0.61065574	0.65573771	0.67581967	0.71721311	0.57377049	0.77459016	0.47131148	0.65163934	0.72131148	0.66393443	0.61885246	0.67213115	0.6352459		
			Val	0.26484018	0.24657534	0.24200913	0.26255708	0.25114155	0.23744292	0.21917808	0.21689498	0.19863014	0.20776256	0.17123288	0.21461187	0.1826484	0.16438356	0.15981735	0.16438356	0.16666667	0.15068493	0.14840183	0.14383562	0.14840183	0.1347032	0.12328767		
	Non1_Acc1_Dat1		Test	0.78278689	0.86065574	0.84016393	0.8442623	0.82786885	0.84016393	0.83606557	0.84016393	0.82786885	0.79918033	0.71721311	0.78278689	0.74180328	0.72540984	0.80737705	0.76639344	0.78639344	0.81557377	0.79098361	0.7704918	0.79508197	0.79508197	0.83196721		
	Acc0_Dat1		Val	0.04950495	0.04950495	0.02970297	0.02475248	0.02475248	0.02475248	0.01980198	0.00990099	0.01980198	0.01485149	0.01485149	0.01485149	0.0049505	0.00990099	0.00990099	0.00990099	0.01485149	0.01485149	0.0049505	0.0049505	0.0049505	0.0049505	0.0049505	
			Test	0.046875	0.03645833	0.04166667	0.03645833	0.05729167	0.06770833	0.046875	0	0.00520833	0.01041667	0	0.015625	0.01041667	0.00520833	0.00520833	0.00520833	0.00520833	0.01041667	0.00520833	0.01041667	0.00520833	0.01041667		
	Non0_Dat1		Val	0.04950495	0.06930693	0.03960396	0.03960396	0.02970297	0.01485149	0.01485149	0	0.00990099	0.0049505	0	0	0.0049505	0.0049505	0	0	0	0	0	0	0	0	0.0049505	
			Test	0.1875	0.25	0.109375	0.13020833	0.09375	0.02604167	0.125	0.02604167	0.02604167	0.03125	0.02083333	0.03645833	0.02604167	0.015625	0.01041667	0.015625	0.015625	0.01041667	0.01041667	0.015625	0.02604167	0.02083333	0.01041667		
	Acc0_Dat1		Val	0.16831683	0.18811881	0.17491749	0.1320132	0.11221122	0.08250825	0.08250825	0.0330033	0.07590759	0.02970297	0.02640264	0.0330033	0.02640264	0.02640264	0.01650165	0.02640264	0.02310231	0.01980198	0.01980198	0.01980198	0.02640264	0.02310231	0.02640264	
			Test	0.02083333	0.01125	0.015625	0.01041667	0.00520833	0.02083333	0.02604167	0.046875	0	0.046875	0.046875	0.01041667	0.02083333	0.05729167	0.00520833	0.00520833	0.01041667	0	0	0.00520833	0.00520833	0.01041667	0.00520833		
	Non0_Dat0_Dat1		Val	0.07260726	0.05280528	0.04290429	0.02970297	0.0330033	0.0330033	0.01980198	0.00330033	0.00990099	0.01980198	0.00660066	0.00660066	0.00990099	0.00660066	0.00660066	0.00990099	0.01320132	0.00660066	0.00330033	0.00990099	0.00990099	0.00660066	0.00990099	
			Test	0.28125	0.19270833	0.11458333	0.05729167	0.078125	0.08333333	0.05208333	0.02604167	0.02083333	0.02083333	0.04166667	0.046875	0.02083333	0.03645833	0.015625	0.015625	0.015625	0.01041667	0.01041667	0.01041667	0.02083333	0.015625	0.02083333		
	Non0_Acc1_Dat1		Val	0.21782178	0.16831683	0.15841584	0.1320132	0.10561056	0.06930693	0.0330033	0.0330033	0.04950495	0.01320132	0.01320132	0.01650165	0.01650165	0.01650165	0.01980198	0.02640264	0.01980198	0.01980198	0.01320132	0.00990099	0.02310231	0.02970297	0.02640264		
			Test	0.21354167	0.3229167	0.234375	0.21875	0.20833333	0.16666667	0.203125	0.11458333	0.16145833	0.05729167	0.05729167	0.015625	0.02083333	0.02083333	0.015625	0.01041667	0.02083333	0.015625	0.01041667	0.015625	0.02083333	0.02083333	0.02083333		
Noun Only			Val	0.26237624	0.21287129	0.21287129	0.19306931	0.26732673	0.21782178	0.18316832	0.16336634	0.14356436	0.16336634	0.16336634	0.16336634	0.16336634	0.13861386	0.14356436	0.12871287	0.13366337	0.12871287	0.11881188	0.07920792	0.08910891	0.09405941	0.08910891		
	Acc0_Dat1		Test	0.08854167	0.08854167	0.06770833	0.078125	0.046875	0.05208333	0.046875	0.046875	0.05208333	0.046875	0.046875	0.046875	0.046875	0.046875	0.19270833	0.140625	0.328125	0.21875	0.3125	0.25	0.328125	0.23958333	0.25	0.16145833	
			Val	0.26732673	0.31683168	0.34158416	0.28217822	0.28217822	0.25742574	0.26732673	0.26732673	0.15841584	0.13861386	0.12871287	0.15346535	0.14851485	0.14356436	0.09405941	0.12871287	0.05445545	0.08415842	0.0990099	0.09405941	0.08415842	0.10891089	0.04950495		
	Non0_Dat1		Test	0.703125	0.703125	0.66145833	0.72395833	0.70833333	0.85416667	0.83333333	0.80729167	0.703125	0.74479167	0.68229167	0.796875	0.80729167	0.78645833	0.80729167	0.828125	0.671875	0.796875	0.8125	0.75520833	0.765625	0.828125	0.73958333		
			Val	0.36633663	0.2871287	0.24092409	0.31353135	0.2970297	0.28052805	0.25082508	0.20132013	0.21122112	0.17161716	0.2310231	0.15511551	0.1650165	0.15841584	0.15181518	0.12871287	0.14851485	0.13531353	0.13531353	0.10561056	0.09570957	0.11221122	0.10561056		
	Acc0_Dat1	Caseless	Test	0.22395833	0.08333333	0.0625	0.09375	0.078125	0.08854167	0.0625	0.03645833	0.05208333	0.04166667	0.05208333	0.05729167	0.04166667	0.05208333	0.05208333	0.01041667	0.015625	0.01041667	0.01041667	0.00520833	0.00520833	0.00520833	0.015625	
			Val	0.33993399	0.3960396	0.38613861	0.40594059	0.33993399	0.23432343	0.28712871	0.24422442	0.21452145	0.26732673	0.19471947	0.18151815	0.19471947	0.14851485	0.1650165	0.16171617	0.1617161	0.1320132	0.14521452	0.12541254	0.14191419	0.10891089	0.11221122		
	Non0_Acc0_Dat1		Test	0.63541667	0.609375	0.796875	0.609375	0.8125	0.76041667	0.85416667	0.71354167	0.81770833	0.64583333	0.69270833	0.66145833	0.703125	0.71354167	0.77083333	0.765625	0.80208333	0.625	0.54166667	0.578125	0.72916667	0.59895833	0.58333333		
			Val	0.28382838	0.25082508	0.23432343	0.2310231	0.22112211	0.21452145	0.21122112	0.17491749	0.1650165	0.13531353	0.14521452	0.17821782	0.14191419	0.12871287	0.10891089	0.12211221	0.10891089	0.09240924	0.09570957	0.09570957	0.10231023	0.08580858	0.07920792		
	Non0_Acc1_Dat1		Test	0.828125	0.6875	0.86979167	0.828125	0.79166667	0.80208333	0.86458333	0.828125	0.88541667	0.88541667	0.765625	0.85416667	0.91666667	0.92708333	0.93229167	0.91145833	0.91666667	0.90625	0.890625	0.91145833	0.88020833	0.91666667	0.91145833		
			Val	0	0.025	0	0.025	0.025	0	0	0.025	0	0	0	0.025	0.025	0	0.025	0.025	0.025	0.025	0	0	0	0	0	
	Acc0_Dat1		Test	0.03846154	0.03846154	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.05769231	0.03846154	0.07692308	0.03846154	0	0.03846154	0.03846154	0.03846154		
			Val	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.025	0.025	
	Non0_Dat1		Test	0.03846154	0.01923077	0.01923077	0.01923077	0.01923077	0.03846154	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.03846154	0.03846154	0.05769231	0.05769231	0.05769231		
			Val	0.13333333	0.1	0.06666667	0.1	0.05	0.1	0.08333333	0.1	0.03333333	0.06666667	0.13333333	0.06666667	0.1	0.08333333	0.08333333	0.05	0.08333333	0.06666667	0.05	0.06666667	0.05	0.06666667	0.1	
	Acc0_Dat1	Cased	Test	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0	0	0	0	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0	0	0.01923077	0	0.01923077	0.01923077	0.03846154	
			Val	0	0.01666667	0.01666667	0.01666667	0.01666667	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01666667	0	0.01666667	0.01666667	0.01666667	
	Non0_Dat0_Dat1		Test	0.03846154	0.03846154	0.01923077	0.01923077	0.03846154	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.03846154	0.03846154	0.03846154	0.05769231	0.03846154	0.03846154	0.03846154	0.03846154		
			Val	0.08333333	0.16666667	0.1	0.13333333	0.13333333	0.08333333	0.08333333	0.06666667	0.08333333	0.08333333	0.08333333	0.08333333	0.1	0.08333333	0.08333333	0.08333333	0.08333333	0.08333333	0.08333333	0.06666667	0.08333333	0.08333333	0.11666667		
	Non0_Acc1_Dat1		Test	0.03846154	0.01923077	0.01923077	0.01923077	0	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077	0.01923077		
Pronoun Only			Val	0.15	0.15	0.15	0.225	0.225	0.15	0.15	0.125	0.15	0.175	0.2	0.125	0.15	0.125	0.125	0.125	0.1	0.175	0.1	0.075	0.1	0.125	0.075	
	Acc0_Dat1		Test	0.25	0.15384615	0.28846154	0.34615385	0.11538462	0.17307692	0.19230769	0.23076923	0.17307692	0.19230769	0.15384615	0.17307692	0.23076923	0.21153846	0.17307692	0.15384615	0.19230769	0.19230769	0.14615385	0.17307692	0.19230769	0.13461538	0.15384615		
			Val	0.25	0.25	0.175	0.25	0.15	0.175	0.15	0.175	0.125	0.25	0.15	0.125	0.225	0.25	0.075	0.075	0.075	0.175	0.075	0.1	0.1	0.075	0.15	
	Non0_Dat1		Test	0.53846154	0.53846154	0.25	0.51923077	0.57692308	0.42307692	0.44230769	0.42307692	0.57692308	0.69230769	0.46153846	0.57692308	0.71153846	0.71153846	0.40384615	0.40384615	0.40384615	0.26923077	0.36538462	0.34615385	0.36538462	0.28846154	0.53846154		
			Val	0.25	0.26666667	0.26666667	0.26666667	0.3	0.25	0.3	0.3	0.23333333	0.28333333	0.25	0.23333333	0.26666667	0.26666667	0.28333333	0.21666667	0.23333333	0.25	0.16666667	0.18333333	0.35	0.16666667	0.21666667		
	Non0_Acc0_Dat1	Caseless	Test	0.09615385	0.09615385	0.15384615	0.11538462	0.23076923	0.21153846	0.25	0.23076923	0.11538462	0.21153846	0.17307692	0.17307692	0.17307692	0.19230769	0.19230769	0.13461538	0.11538462	0.15384615	0.15384615	0.11538462	0.21153846	0.09615385	0.11538462		
			Val	0.2	0.21666667	0.25	0.18333333	0.2	0.25	0.18333333	0.21666667	0.18333333	0.15	0.15	0.15	0.15	0.11666667	0.11666667	0.11666667	0.13333333	0.11666667	0.15	0.13333333	0.11666667	0.2	0.16666667	0.16666667		
	Non0_Acc1_Dat1		Test	0.25	0.34615385	0.34615385	0.28846154	0.25	0.53846154	0.51923077	0.25	0.25	0.44230769	0.42307692	0.32692308	0.40384615	0.38461538	0.38461538	0.40384615	0.21153846	0.36538462	0.38461538	0.30769231	0.13461538	0.36538462	0.19230769		
			Val	0.31666667	0.33333333	0.31666667	0.31666667	0.3	0.3	0.35	0.3	0.33333333	0.26666667	0.35	0.36666667	0.33333333	0.33333333	0.36666667	0.33333333	0.3	0.31666667	0.3	0.31666667	0.31666667	0.28333333	0.3	
	Non1_Acc0_Dat1		Test	0.19230769	0.40384615	0.19230769	0.36538462	0.36538462	0.36538462	0.63461538	0.36538462	0.63461538	0.48076923	0.63846615	0.63461538	0.61538462	0.63461538	0.61538462	0.59615385	0.59615385	0.59615385	0.59615385	0.57692308	0.32692308	0.55769231		
			Val	-																							
	Acc_random		Test	0.479508	0.52868852	0.5	0.3770491	0.52459016	0.40163934	0.35245902	0.46311475	0.43442623	0.45901639	0.46311475	0.59836066	0.52459016	0.34836066	0.6557377	0.5204918	0.55737705	0.6352459	0.53688525	0.58606557	0.52459016	0.63114754	0.5204918	
			Val	-																							
	Non0_random	Cased	Test	0.34016393	0.48770492	0.50409836	0.5	0.57377049	0.44262295	0.56147541	0.45491803	0.34016393	0.4795082	0.47131148	0.50409836	0.3647541	0.50819672	0.47131148	0.62295082	0.56147541	0.56557377	0.57377049	0.43032787	0.45081967	0.44262295		
			Val	-																							
	All Random		Test	0.36885246	0.48770492	0.43032787	0.44262295	0.35655738	0.30327869	0.2295082	0.57377049	0.42622951	0.41393443	0.40163934	0.46311475	0.55737

D Full Word2Vec Results

Random	**Cased**	**Acc0_Dat1**	**Val**	0.01662049861
			Test	0.01739130435
		Nom0_Dat1	**Val**	0.01333333333
			Test	0
		Nom1_Acc0_Dat1	**Val**	0.1423357664
			Test	0.06086956522
		Nom0_Acc0_Dat1	**Val**	0.02737226277
			Test	0.01739130435
		Nom0_Acc1_Dat1	**Val**	0.147810219
			Test	0.05652173913
	Caseless	**Acc0_Dat1**	**Val**	0.1982758621
			Test	0.3051643192
		Nom0_Dat1	**Val**	0.2841225627
			Test	0.4084507042
		Nom1_Acc0_Dat1	**Val**	0.2129277567
			Test	0.2957746479
		Nom0_Acc0_Dat1	**Val**	0.2585551331
			Test	0.338028169
		Nom0_Acc1_Dat1	**Val**	0.319391635
			Test	0.3427230047
Same Token Amount	**Cased**	**Acc0_Dat1**	**Val**	0.03370786517
			Test	0.02173913043
		Nom0_Dat1	**Val**	0.007272727273
			Test	0
		Nom1_Acc0_Dat1	**Val**	0.1523341523
			Test	0.05217391304
		Nom0_Acc0_Dat1	**Val**	0.03194103194
			Test	0.01304347826
		Nom0_Acc1_Dat1	**Val**	0.1646191646
			Test	0.04782608696
	Caseless	**Acc0_Dat1**	**Val**	0.1984126984
			Test	0.3051643192
		Nom0_Dat1	**Val**	0.2996108949
			Test	0.3849765258
		Nom1_Acc0_Dat1	**Val**	0.2663185379
			Test	0.2723004695
		Nom0_Acc0_Dat1	**Val**	0.2480417755
			Test	0.3568075117
		Nom0_Acc1_Dat1	**Val**	0.3446475196
			Test	0.3990610329

Table 5: Full results for Word2Vec model. Including the test error rates for the control datasets (Same Token Amount, Noun Only, and Pronoun Only) and test error rate for the control probe. The table also includes all error rates for the validation set, minus the control probe where there was no validation set. The results reported in the main paper are from the Random dataset.

Noun Only	**Cased**	**Acc0_Dat1**	**Val**	0.02312138728
			Test	0.0393258427
		Nom0_Dat1	**Val**	0.01149425287
			Test	0.01685393258
		Nom1_Acc0_Dat1	**Val**	0.2432432432
			Test	0.06179775281
		Nom0_Acc0_Dat1	**Val**	0.03861003861
			Test	0.02808988764
		Nom0_Acc1_Dat1	**Val**	0.2007722008
			Test	0.0393258427
	Caseless	**Acc0_Dat1**	**Val**	0.2075471698
			Test	0.2795031056
		Nom0_Dat1	**Val**	0.28125
			Test	0.3229813665
		Nom1_Acc0_Dat1	**Val**	0.268907563
			Test	0.2670807453
		Nom0_Acc0_Dat1	**Val**	0.256302521
			Test	0.3602484472
		Nom0_Acc1_Dat1	**Val**	0.3655462185
			Test	0.3229813665
Pronoun Only	**Cased**	**Acc0_Dat1**	**Val**	0
			Test	0
		Nom0_Dat1	**Val**	0
			Test	0.01923076923
		Nom1_Acc0_Dat1	**Val**	0.1166666667
			Test	0
		Nom0_Acc0_Dat1	**Val**	0
			Test	0.01923076923
		Nom0_Acc1_Dat1	**Val**	0.15
			Test	0.01923076923
	Caseless	**Acc0_Dat1**	**Val**	0.1052631579
			Test	0.25
		Nom0_Dat1	**Val**	0.175
			Test	0.25
		Nom1_Acc0_Dat1	**Val**	0.2586206897
			Test	0.09615384615
		Nom0_Acc0_Dat1	**Val**	0.2068965517
			Test	0.25
		Nom0_Acc1_Dat1	**Val**	0.3275862069
			Test	0.1923076923
Control		**Acc_random**	**Val**	
			Test	0.4
	Cased	**Nom random**	**Val**	
			Test	0.5347826087
		All Random	**Val**	
			Test	0.5217391304
	Caseless	**Acc_random**	**Val**	
			Test	0.3896713615
		Nom random	**Val**	
			Test	0.4835680751
		All Random	**Val**	
			Test	0.4882629108

E Dative Predicates

Benefactive/Malefactive Verbs: *helfen* 'to help', *danken* 'to thank', *gratulieren* 'to congratulate', *folgen* 'to follow', *glauben* 'to believe', *zuhören* 'to listen', *vertrauen* 'to trust', *antworten* 'to answer', *verzeihen* 'to forgive', *drohen* 'to threaten', *widersprechen* 'to contradict', *zustimmen* 'to agree', *raten* 'to recommend', *schaden* 'to harm', *dienen* 'to serve', *begegnen* 'to respond', *zureden* 'to persuade', *nachlaufen* 'to chase', *ausweichen* 'to avoid', *beisteuern* 'to contribute', *ermöglichen* 'to enable'

Ditransitives: *zuordnen* 'to allocate', *zuteilen* 'to allocate, *zuwenden* 'to bestow', *zustecken* 'to slip somebody something', *vermitteln* 'to impart', *verkaufen* 'to sell', *verleihen* 'to loan', *verdanken* 'to owe', *vermachen* 'to leave somebody something, *hinzufügen* 'to add somebody to something', *darbieten* 'to offer somebody something', *geben* 'to give', *schenken* 'to give', *bescheren* 'to give', *(ein)bringen* 'to bring', *unterstellen* 'to leave somebody something', *unterlegen* 'to underlay something with something', *überlassen* 'to transfer', *übergeben* 'to hand somebody something', *überreichen* 'to present somebody with something', *nachempfinden* 'to base something on somebody', *näherbringen* 'to make something more accessible to somebody', *entgegensetzen* 'to counter something with something', *entgegenkommen* 'to make concessions to someone', *erschweren* 'to impede', *weihen* 'to dedicate', *verschreiben* 'to prescribe', *erzählen* 'to tell somebody something', *empfehlen* 'to recommend', *versprechen* 'to promise', *mitteilen* 'to disclose', *anvertrauen* 'to entrust somebody with something', *vorwerfen* 'to accuse somebody of something', *goennen* 'to begrudge something something', *platzmachen* 'to make room', *raten* 'to recommend', *wünschen* 'to want something', *zeigen* 'to explain', *stehlen* 'to steal', *rauben* 'to steal', *entziehen* 'to deprive', *entreißen* 'to steal', *entwenden* 'to steal', *wegnehmen* 'to take something away from someone', *abnehmen* 'to take something away from someone', *vorenthalten* 'to keep something from somebody', *verweigern* 'to refuse', *verschweigen* 'to hide something from someone', *verheimlichen* 'to conceal something from someone'

Stimulus-Experiencer Verbs: *einfallen* 'to think of', *fehlen* 'to miss', *gefallen* 'to enjoy', *leidtun* 'to be sorry', *passen* 'to work/be good for someone', *anpassen* 'to fit', *schmecken* 'to like', *ergehen* 'to indulge', *ist + DP ADJ* 'somebody is something'

F Control Probes

A criticism of linguistic probing is that classifier may be learning the task and not actually showing if the language model has encoded the information. A solution to see if the classifier is just solving the task is to include a control probe (Hewitt and Liang, 2019). These kind of probes are trained on a random organisation of the training data; for us that means class 0 and class 1 being made up of a random selection of all the cases. We included three control probes: *Acc Control Probe* (class 0 = random accusatives and random datives; class 1 = random accusatives and random datives), *Nom Control Probe* (class 0 = random nominatives and random datives; class 1 = random nominatives and random datives), and *All Control Probe* (class 0 = random nominatives, random accusatives, and random datives; class 1 = random nominatives, random accusatives, and random datives). The architecture of these probes and training is the same from the classifiers in the main paper. For each model, we trained a control probe on the random training set (i.e. not on any of the control datasets), and are presenting the control probes that were trained on the layers that were presented in the paper (e.g. for *BERT* layer 11, for *mBERT* layer 10, for *LLäMmlein 1B* layer 21). We then classified the test datives. We have assumed classifying the datives as 1 as the correct answer (of course, given the organisation of the training data, this is an arbitrary assignment). We did this for the cased condition (Figure 5) and the caseless condition (Figure 6).

For the majority of the probes and models in the cased condition (Figure 5), they are at chance or around chance. However, there is a notable exception with *BERT* with the *All Probe*. For the caseless condition (Figure 6), there is once again a noticeable difference with *BERT* as well as *mBERT*. Given that we know *mBERT* was not good at the task in the main paper, we will put those controls aside to focus on *BERT*. When looking closer at the probabilities the control probe has with *BERT* in the cased *All Probe*, the caseless *Acc Probe* and *All Probe*, we can see that for both class 0 and class 1 the probabilities are close to 0.5; this is in contrast with the respective probes in the main experiments. More specifically, in the caseless experiment, *BERT* scored a similar error rate in the

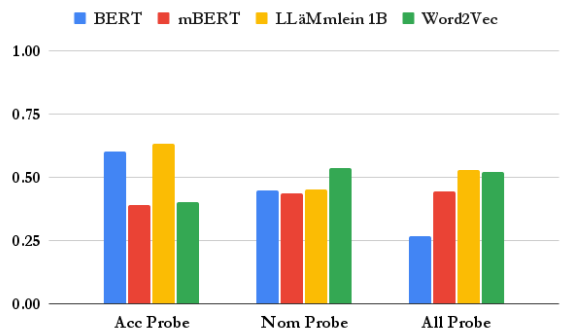


Figure 5: Control probes on the cased representation. Error rates for *BERT* layer 11, *mBERT* layer 10, *LLäMmlein 1B* layer 21, and *Word2Vec* baseline.

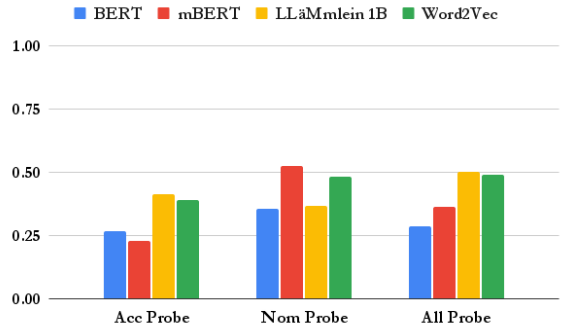


Figure 6: Control probes on the decased representation. Error rates for *BERT* layer 11, *mBERT* layer 10, *LLäMmlein 1B* layer 21, and *Word2Vec* baseline.

Nom/Acc ∼ *Dat* data configuration (Figure 4) to the caseless *All Probe* (Figure 6), however, when we look at the probabilities for class 0 and class 1 on the dative test set, we see a stark difference. The experimental probe (Figure 7) skews to assigning class 1 probabilities as around 0.9, whilst the control probe assigns class 1 around 0.5. What we can take away is that although control probes look like they are solving their task, the probabilities show that they are assigning almost a random chance for either class.

There is a strategy with categorising nominative, accusative, and dative arguments whereby you can utilise argument type information, as seen with the effect of the *pronoun only* and *noun only* datasets and how the *Word2Vec* model was able to do the caseless task relatively well with zero context and with the exact same arguments in all cases/classes (e.g. the caseless pronoun representations). Considering that the *BERT* probes in the *Acc Probe* and *All Probe* in the caseless condition and the *All Probe* in the cased condition are around an error rate of

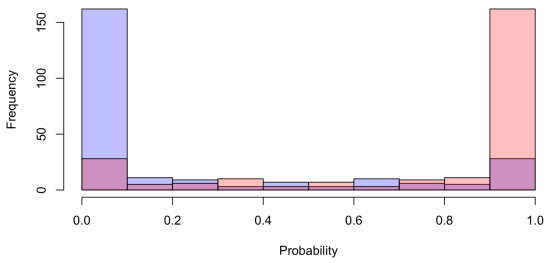


Figure 7: The probabilities for class 0 (blue) and probabilities for class 1 (red) for *BERT* in the caseless *Nom/Acc* ∼ *Dat* condition, on the dative test set.

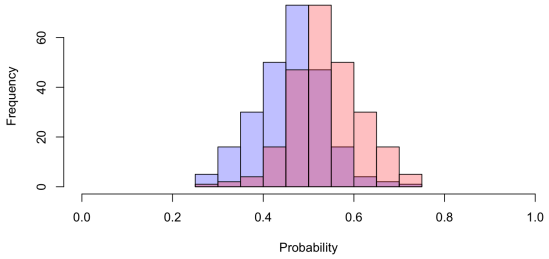


Figure 8: The probabilities for class 0 (blue) and probabilities for class 1 (red) for *BERT* in the caseless *All probe*, on the dative test set.

30% and this is the same of the *Word2Vec* model in the caseless experiment (section 5), it may be that these control probes are training on some sort of argument distribution to achieve the task. Given the probes being close to the *Word2Vec* caseless baseline, the probe's probabilities of class 0 and class 1 both being close to 50%, and the fact that *BERT* performed better in the caseless *Acc* ∼ *Dat* and *Nom/Dat* ∼ *Acc* condition compared to the *Acc Probe* and *All Probe* respectively, and performed worse in the *Nom* ∼ *Acc/Dat* compared to the *All Probe*, we can conclude that the non-control probes are not training on argument type distributions, but that this information can influence the classifiers, something that we state in our discussion.

G Frequency Effects

To test the impact of frequency, we conducted a spearman's Rank Correlation to see if there is a correlation between an argument's frequency and the probability of being assigned class 1. We only looked at *BERT* (as it was the model that performed the best) and the dative test set in the caseless condition (section 5), and we did a test for each data con-

figuration. For DPs, we only used the frequency of the nouns. We obtained the argument frequencies from the *Decow* dataset (Schäfer, 2015; Schäfer and Bildhauer, 2012), if a word was not present it was assigned a frequency of 0. For *Acc* $\sim$ *Dat*, $r_s(224) = -.282$, $p < 0.05$. For *Nom* $\sim$ *Dat*, $r_s(224) = -.189$, $p < 0.05$. For *Nom/Dat* $\sim$ *Acc*, $r_s(224) = -.323$, $p < 0.05$. For *Nom/Acc* $\sim$ *Dat*, $r_s(224) = -.206$, $p < 0.05$. For *Nom* $\sim$ *Acc/Dat*, $r_s(224) = -.351$, $p < 0.05$. Overall, each data configuration has a weak negative correlation between probability and frequency that is statistically significant. Given that it is weak, we conclude that frequency is not overly impactful to the classifiers probability.

H Pondering on Ergativity

An anonymous reviewer wondered how the impact of German being an nominative-aligned language affected the results in comparison to an ergative-aligned language. Here, a nominative language is one where the transitive subject and intransitive subject are morphologically[9] marked as the same. Whilst an ergative language is one where the intransitive subject and the transitive object are morphologically marked the same. Although it would be easy to remark that the experiment should just be re-run but on an ergative language, there are some particular difficulties with this: (i) we lack resources for ergative languages (see Papadimitriou et al. (2021) for a discussion on their difficulty on finding enough resources for ergative languages), (ii) ergative languages tend to have split systems (i.e. they are not always consistently ergative (Dixon, 1994)), and (iii) as ergative case is a dependent case (Baker, 2015), if you have an inherent dative object, the transitive subject should be marked as absolutive not ergative. That is by no means to say we should not be investigating language models' case representations of ergative languages (we absolutely should), it is to say it would be difficult to adapt *this* specific experiment to an ergative language, as it was designed specifically with German in mind. However, we can ponder upon how a language model may differently represent inherent cases in ergative languages.

Given that language models like *mBERT* encode intransitive subjects as objects in ergative languages (Papadimitriou et al., 2021), we may find that in an ergative language with an inherent dative on objects, the active dative and passivised dative will be classified as the same when the overt case markings are removed - obviously this depends on if a language model analyses passivised subjects as intransitive subjects (which are then analysed as 'objects'). The other interesting caveat concerns the dependent nature of ergative case; if an inherent dative object existed in this ergative language, then the absolutive case and dative case would not be in a complementary distribution like the accusative and dative in German, given that the normally ergative transitive subject would become absolutive when paired with a dative argument. However, the dative would be in a complementary distribution with the ergative case. This actually matches with the German, as the complementary distribution is between dependent and inherent cases, of which accusative and ergative are. So, we would expect some sort of encoding case split between ergative and an inherent dative. Presumably then, a language model would compute the case of the 'object' first, and if there is said X of verbs then dative would be picked over absolutive. However, unlike the German language model, there would need to be some extra information for absolutive to come about on the transitive subject, instead of predicting the ergative case. Of course, this is just a hypothesis, and an undeveloped one at that given this is not encoding the split nature of ergative systems. However, ergative case representation should absolutely be investigated with reference to this question, but with an experiment that is specifically designed for it.

[9] We are ignoring syntactic ergativity and accusativity

A Logical Analysis of Autosegmental Approaches to Root-and-Pattern Morphology in Arabic

Sarah Payne
Stony Brook University
sarah.payne@stonybrook.edu

Abstract

Autosegmental approaches to Arabic root-and-pattern morphology generally take a three-tier approach, with tiers corresponding to the prosodic template, consonantal root, and affixes (e.g., McCarthy, 1981); association between these tiers proceeds from left-to-right. However, Jardine (2017) shows that left-to-right association exceeds regular computation for autosegmental representations of arbitrary length, challenging the cognitive plausibility of this approach. This paper demonstrates that in the case of Arabic morphology, the constraints of the system itself — in particular, the finite length of the consonantal root — allow such a left-to-right autosegmental association to not only be definable with Monadic Second Order (MSO) logic, but with First Order logic. This paper introduces a logical relational structure formalizing the three-tier autosegmental representations and defines a set of transductions which apply in parallel over these structures to yield well-formed root and affix associations.

1 Introduction

Semitic languages, including Arabic, are known for their derivational "root-and-pattern" morphology in which a single, typically triconsonantal, root is associated with various prosodic patterns to yield semantically related stems. Previous work has accounted for root-and-pattern morphology in the framework of Autosegmental Phonology (Goldsmith, 1976; Clements and Ford, 1979; McCarthy, 1981; Clements and Keyser, 1983, i.a.). Under such accounts, root consonants are taken as the autosegmental melody to be associated to the melody-bearing consonantal elements of the prosodic template with a left-to-right association. This left-to-right association is evidenced by forms such as *KTaaBaB* from the consonantal root *KTB*, in which the final consonant of the root fills both the third and fourth position in the prosodic template.

A great deal of work has demonstrated that patterns in natural language belong to the class of *regular* languages or some *subregular* subset of these (Chomsky, 1957, 1963; Kaplan and Kay, 1994; Heinz, 2018; Graf, 2022, i.a.). The regular class of languages is known to be equal in expressive power to Monadic Second-Order logic (MSO) on finite words; thus, we expect natural language patterns to be adequately captured by MSO logic (Büchi, 1960). However, Jardine (2017) shows that left-to-right association of autosegmental representations of arbitrary length is not definable with MSO logic, and thus is not regular. This computational complexity challenges the cognitive plausibility of the autosegmental approach, particularly when contrasted with the subregular nature of other morphophonological patterns.

In this paper, I demonstrate that the constraints of the Arabic root-and-pattern system — in particular, the finite length of the root — allow us to define transductions over autosegmental representations using only First Order logic, a subset of MSO. The present work focuses only on triconsonantal roots, but the analysis can and should be expanded for roots of other lengths (§4). This paper proceeds as follows: in §2, I introduce the Arabic data and the autosegmental analysis of McCarthy (1981). In §3.1, I define the relational model of autosegmental representations which we will use in our transductions, and in §3.2 I define the set of logical transductions used to capture the Arabic derivational morphological system as analyzed by McCarthy. Finally, §3.3 illustrates these representations and transductions with a concrete example, §4 discusses these results, and §5 concludes.

2 Root & Pattern Morphology in Arabic

Table 1 gives the Modern Standard Arabic data analyzed by McCarthy (1981): the fifteen *binyanim* of the root *KTB*, where *binyanim* is a term bor-

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 90–99
July 3-4, 2026 ©2026 Association for Computational Linguistics

	Perfective		Imperfective		Participle	
Binyan	Active	Passive	Active	Passive	Active	Passive
I	KaTaB	KuTiB	aKTuB	uKTaB	KaaTiB	maKTuuB
II	KaTTaB	KuTTiB	uKaTTiB	uKaTTaB	muKaTTiB	muKaTTaB
III	KaaTaB	KuuTiB	uKaaTiB	uKaaTaB	muKaaTiB	muKaaTaB
IV	ʔaKTaB	ʔuKTiB	uʔaKTiB	uʔaKTaB	muʔaKTiB	muʔaKTaB
V	taKaTTaB	tuKuTTiB	ataKaTTaB	utaKaTTaB	mutaKaTTiB	mutaKaTTaB
VI	taKaaTaB	tuKuuTiB	ataKaaTaB	utaKaaTaB	mutaKaaTiB	mutaKaaTaB
VII	nKaTaB	nKuTiB	anKaTiB	unKaTaB	munKaTiB	munKaTaB
VIII	KtaTaB	KtuTiB	aKtaTiB	uKtaTaB	muKtaTiB	muKtaTaB
IX	KTaBaB		aKTaBiB		muKTaBiB	
X	staKTaB	stuKTiB	astaKTiB	ustaKTaB	mustaKTiB	mustaKTaB
XI	KTaaBaB		aKTaaBiB		muKTaaBiB	
XII	KTawTaB		aKTawTiB		muKTawTiB	
XIII	KTawwaB		aKTawwiB		muKTawwiB	
XIV	KTanBaB		aKTanBiB		muKTanBiB	
XV	KTanBay		aKTanBiy		muKTanBiy	

Table 1: The root *KTB* "write" in all fifteen binyanim, taken from McCarthy (1981); gaps indicate binyanim that are regularly intransitive and thus not susceptible to passivization. Root consonants are indicated with capitalization.

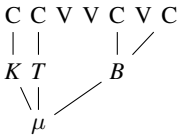
rowed by McCarthy from Hebrew morphology for the classification of derived stems from a given root. Throughout the paper, root consonants are capitalized to clearly separate them from affixes.

Under McCarthy's analysis, which is couched in the framework of Autosegmental Phonology (Goldsmith, 1976; Clements and Ford, 1979; Clements and Keyser, 1983), each form in Table 1 is derived by associating the given triconsonantal root with a melody of consonantal slots and vowels corresponding to the given binyan, as well as affixes in some cases. The prosodic template of a given binyan corresponds to the *segmental* level of the autosegmental analysis, while the root composes the *melody* on a single, morphologically-defined tier. McCarthy analyzes affixes — consonant(s) appearing in a given binyan which are not a part of the root — on a separate autosegmental tier, specified for the same features as the root consonants. The role of the phonological apparatus under this analysis, then, is to take in underlying unassociated elements of these three tiers and output a surface form with well-formed associations between them.

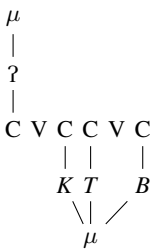
McCarthy defines this prosodic template solely in terms of the features ±Segmental and ±Syllabic, with the relevant combinations being abbreviated as C ([+Segmental, -Syllabic]) and V ([+Segmental, +Syllabic]). I do not make a commitment as to whether vowel features are represented in the prosodic template or separately, as it would have little bearing on the analysis I present in §3.2; see McCarthy for discussion. Under McCarthy's analysis, the consonantal root takes as melody-bearing elements the [-Syllabic] positions of the prosodic template of the given binyan. The consonants of the root and any affixes are defined in terms of all relevant phonological features except ±Segmental and ±Syllabic, though they are often represented as segments for ease of exposition. For example, the *K* in (1) indicates the bundle {Dorsal, -Voice, -Continuant, -Sonorant, ...} save any specification for Segmental or Syllabic.

McCarthy assumes that material on the affixal tier is associated with the prosodic template *before* material on the root tier, so that the root tier material may then be associated from left-to-right with the remaining slots on the prosodic template, following the universal constraints of Clements and Ford (1979). This left-to-right association is evidenced in forms such as *KTaaBaB*, as shown in (1). To illustrate the affixation process, the proposed autosegmental representation of *ʔaKTaB*, which contains the prefix [ʔ-], is given in (2) .

(1) The autosegmental representation of *KTaaBaB* proposed by McCarthy.



(2) The autosegmental representation of *ʔaKTaB* proposed by McCarthy.



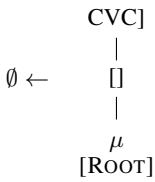
It is generally sufficient to associate the affix with the first consonant on the prosodic template, following the universal left-right association proposed by Clements and Ford. The reflexive [t], however, is prefixed in binyan V (***t-****aKaTTaB*) and VI (***t-****aKaaTaB*), but infixed in binyan VIII (***K-t-****aTaB*). To account for this, McCarthy proposes a "flop" rule which switches the association of the affix to the second slot in binyan VIII (3). This rule applies after the association of the affix, but before the association of the root, so that the root can associate left-to-right with the remaining available slots. For example, *KtaTaB* is derived by first associating the affix [t] with the CCVCVC template and applying the flop rule, so that when the root *KTB* is associated, *K* will fill the empty first consonant slot, *T* will fill the next empty consonant slot, which is the third, and *B* will fill the final slot.

(3) McCarthy's binyan VIII flop rule.

In addition to [t], [-y] in binyan XV is purely suffixing, while [-n-] and [-w-] in binyanim XII-XV are infixing. Because they are *never* prefixed, McCarthy argues that there is no reason to posit a flop rule of the form of (3) for these binyanim, but rather that the rules of association themselves must indicate where these affixes are to be associated to the prosodic template. It is straightforward to see that this approach could also be applied to binyan VIII rather than the flop rule, and this is what we will do in §3.2.1. Though one could argue that such an analysis of binyan VIII loses the insight that [t-] is a prefix in the other two binyanim in which it appears, either method requires a binyan-specific rule, whether we only apply (3) in binyan VIII or only associate [t-] directly with the second slot in this binyan. This insight is crucial not only for the transductions we define in §3.2.1, but also for the representations in §3.1.

Finally, McCarthy addresses the gemination of the medial root consonant in binyanim II, V, and XIII (e.g., *KaTTaB*), with an "erasure" rule (4) which erases the association between the final root consonant and the penultimate consonantal slot; the penultimate C is then re-associated with the autosegement associated with the slot to its left. Erasure is also a *binyan-specific* rule.

(4) McCarthy's erasure rule.



Having reviewed the autosegmental analysis of McCarthy (1981), we now turn to a formalization of this approach in terms of logical transductions. Similarly to the phonological apparatus under the autosegmental approach, these transductions take in unassociated roots, prosodic templates, and affixes, and return a surface form with well-formed associations between elements on these three tiers.

3 Logical Formalization

3.1 Representations

Before defining our logical transductions, we must first define the representations that they will operate over. I build on the 2-tier relational model for autosegmental representations proposed by Jardine (2024) to give a relational model with three tiers: (1) the prosodic template, (2) affixes, and (3) the consonantal root. While Jardine (2024) uses the successor ($\lhd$) relation, I use the precedence ($<$) relation, allowing us to reference only the order of consonantal slots in the prosodic template (§3.2) and to capture the fact that although Arabic derivational morphology preserves the overall *order* of the consonants in the root, it does not preserve strict adjacency.

3.1.1 Model Signature

Our model $\mathfrak{R}$ is similar to the precedence string model, with the addition of two further relations: $\circ_a$ defines the association between affixes and the prosodic template, and $\circ_r$ defines the association between the root and the prosodic template:

$$\mathfrak{R} = \{<, \circ_a, \circ_r\} \cup \mathcal{P} \cup \mathcal{C} \cup \mathcal{R} \cup \mathcal{M} \tag{1}$$

$\mathcal{P}$ is the set of features (unary relations) relevant for defining the prosodic template, $\mathcal{R}$ the set relevant to the roots, $\mathcal{M}$ the set relevant to the morphological affixes, and $\mathcal{C}$ the set of phonological features for consonants on both the root and affix tiers.

Following McCarthy (1981), $\mathcal{P}$ contains the features `segm` (corresponding to McCarthy's ±SEGMENTAL) and `syll` (corresponding to McCarthy's ±SYLLABIC). It also includes unary features for each of the 15 binyanim (i.e., I, II, etc); because processes such as the flop (3) and erasure (4)

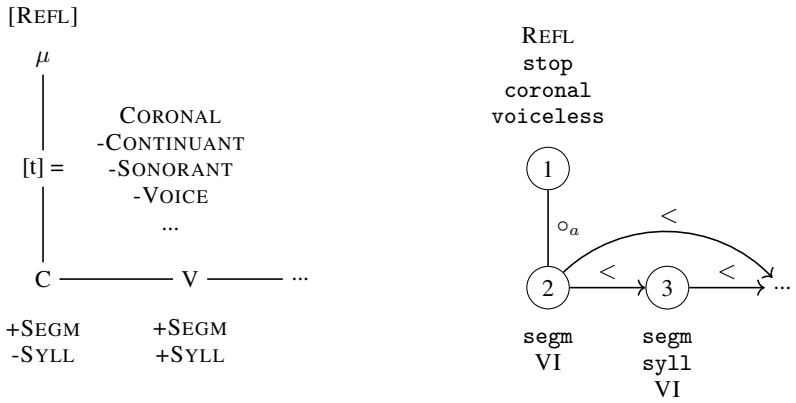


Figure 1: The original McCarthy autosegmental representation of [t-] prefix attached to the binyan VI (left) and our proposed logical representation of the same prefix on the same binyan (right)

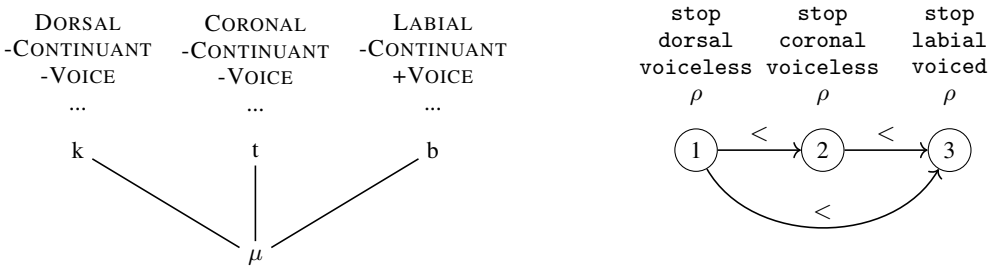


Figure 2: The original McCarthy autosegmental representation of the root *KTB* (left) and our proposed logical representation of the same root (right).

rules apply only to some binyanim, we must know which binyan we are forming to know whether the corresponding transductions (§3.2) apply. If we wished for vowels to be fully-specified in the prosodic template, we could include features such as `High`, `Round`, etc. in $\mathcal{P}$. However, whether or not these features are included has no bearing on the transductions we define in §3.2, so I do not make any theoretical claims regarding this issue.

$\mathcal{C}$ contains phonological features for consonants on the root and affix tiers, including information about place (e.g., `dorsal`), manner (e.g., `stop`) and voicing (e.g., `voiceless`), while $\mathcal{M}$ and $\mathcal{R}$ contain features relevant for specifying elements on the affix or root tiers, respectively, and differentiating between them. Under this approach, elements are on the affixal tier if they bear some feature in $\mathcal{M}$, and on the root tier if they bear some feature in $\mathcal{R}$; this aids in defining constraints on valid representations in §3.1.2.

$\mathcal{M}$ contains the morphosyntactic features (e.g., `Reflexive`) for affixes. Though McCarthy often uses μ to indicate elements on both tiers, he adds morphosyntactic features such as REFLEXIVE in his representation of the [t-] affix in the flop rule (3) to define affixal elements. While McCarthy represents features such as REFLEXIVE and μ on yet *another* autosegmental tier, I collapse these to one tier; this approach is illustrated in Figure 1 for the [t-] prefix in binyan VI. $\mathcal{R}$ contains a feature ρ which holds only for elements of morphological roots; this is analagous to the ROOT feature used by McCarthy in his definition of the erasure rule (4). The logical representation of the root *KTB* is given in Figure 2.

3.1.2 Constraints on Valid Representations

Having given the model signature of our three-tier autosegmental model, we wish to define which $\mathfrak{R}$-structures conforming to this signature are valid autosegmental representations. We begin by defining constraints on well-formed tiers. Firstly, elements in the prosodic template must bear only features which are in $\mathcal{P}$, so bearing some feature in $\mathcal{P}$ precludes bearing any features in $\mathcal{M}, \mathcal{C}$ or $\mathcal{R}$:

$$\texttt{sep}_{\texttt{p}} \stackrel{\text{def}}{=} p(x) \rightarrow (\forall c \in \mathcal{C})[\neg c(x)] \wedge (\forall m \in \mathcal{M})[\neg m(x)] \wedge (\forall r \in \mathcal{R})[\neg r(x)] \quad (2)$$

Similarly, segments on the affix tier and root tier must bear both consonantal features and *either* root or affix features, with one precluding the other:

$$\texttt{sep}_{\texttt{m}} \stackrel{\text{def}}{=} m(x) \rightarrow (\exists c \in \mathcal{C})[c(x)] \wedge (\forall p \in \mathcal{P})[\neg p(x)] \wedge (\forall r \in \mathcal{R})[\neg r(x)] \quad (3)$$

$$\texttt{sep}_\texttt{r} \stackrel{\text{def}}{=} r(x) \rightarrow (\exists c \in \mathcal{C})[c(x)] \quad (4)$$
$$\wedge\, (\forall p \in \mathcal{P})[\neg p(x)]$$
$$\wedge\, (\forall m \in \mathcal{M})[\neg m(x)]$$

We can formalize this separation of affix, root, and prosodic template elements as follows:

$$\texttt{sep} \stackrel{\text{def}}{=} \bigwedge_{r\in\mathcal{R},p\in\mathcal{P},m\in\mathcal{M}} \texttt{sep}_\texttt{p} \quad (5)$$
$$\wedge\ \texttt{sep}_\texttt{m} \wedge \texttt{sep}_\texttt{r}$$

We now define whether an element is a prosodic template slot, affixal segment, or root segment:

$$\texttt{p_slot}\ (x) \stackrel{\text{def}}{=} \bigvee_{p\in\mathcal{P}} p(x) \quad (6)$$

$$\texttt{affix}\ (x) \stackrel{\text{def}}{=} \bigvee_{m\in\mathcal{M}} m(x) \quad (7)$$

$$\texttt{root}\ (x) \stackrel{\text{def}}{=} \bigvee_{r\in\mathcal{R}} r(x) \quad (8)$$

Using these predicates, we can determine whether any two elements are on the same tier:

$$\texttt{same_tier}\ (x,y) \stackrel{\text{def}}{=} (\ \texttt{p_slot}\ (x) \quad (9)$$
$$\wedge\ \texttt{p_slot}\ (y))$$
$$\vee (\ \texttt{affix}\ (x) \wedge \texttt{affix}\ (y))$$
$$\vee (\ \texttt{root}\ (x) \wedge \texttt{root}\ (y))$$

Well-formed tiers are those in which all elements are of the same type:

$$\texttt{WFT} \stackrel{\text{def}}{=} (\forall x, y) \quad (10)$$
$$[(x < y) \rightarrow \texttt{same_tier}\ (x,y)$$
$$\wedge\, (x = y) \rightarrow \texttt{same_tier}\ (x,y)]$$

Next, we define well-formed associations between affixes and prosodic templates and between roots and prosodic templates. We begin by defining the consonantal slots in the prosodic template:

$$\texttt{c_slot}\ (x) \stackrel{\text{def}}{=} \texttt{p_slot}\ (x) \wedge \neg\texttt{syll}(x) \quad (11)$$

For affixes, we require an association between a C slot and an affixal element, while for roots, we require an association between a consonant slot and a root element.

$$\texttt{WFA}_\texttt{a}\ (x,y) \stackrel{\text{def}}{=} \texttt{c_slot}\ (y) \quad (12)$$
$$\wedge\ \texttt{affix}\ (x) \wedge x \circ_a y$$

$$\texttt{WFA}_\texttt{r}\ (x,y) \stackrel{\text{def}}{=} \texttt{c_slot}\ (y) \quad (13)$$
$$\wedge\ \texttt{root}\ (x) \wedge x \circ_r y$$

Well-formed associations require that a single C slot only associate with a single element, either from a root or affix. Notably, the converse isn't required: a single root or affix consonant can occupy multiple C slots on the prosodic template.

$$\texttt{WFA} \stackrel{\text{def}}{=} (\forall x, y)[x \circ_a y \rightarrow (\ \texttt{WFA}_\texttt{a}\ (x,y) \quad (14)$$
$$\wedge\, (\nexists z)[z \circ_a y \wedge z \neq x]$$
$$\wedge\, (\nexists w)[w \circ_r y])$$
$$\wedge\, x \circ_r y \rightarrow (\ \texttt{WFA}_\texttt{r}\ (x,y)$$
$$\wedge\, (\nexists z)[z \circ_r y \wedge z \neq x]$$
$$\wedge\, (\nexists w)[w \circ_a y])]$$

Finally, I provide an adaptation of the No-Crossing Constraint predicate defined by Jardine (2024):

$$\texttt{NCC} \stackrel{\text{def}}{=} (\forall x_1, x_2, y_1, y_2)[(x_1 \circ_a x_2 \quad (15)$$
$$\wedge\, y_1 \circ_a y_2 \wedge x_1 < x_2) \rightarrow x_2 < y_2$$
$$\wedge\, (x_1 \circ_r x_2 \wedge y_1 \circ_r y_2 \wedge x_1 < x_2)$$
$$\rightarrow x_2 < y_2]$$

From the above predicates, we define a predicate that determines whether an $\mathfrak{R}$-structure of the form in (1) is a valid autosegmental representation:

$$\texttt{ASR} \stackrel{\text{def}}{=} \texttt{sep} \wedge \texttt{WFT} \wedge \texttt{WFA} \wedge \texttt{NCC} \quad (16)$$

It is straightforward to see that the $\mathfrak{R}$-structures in Figures 1 and 2 are both valid autosegmental representations according to this predicate.

3.2 Logical Transductions

Having defined our relational structures, we can now define the set of logical transductions which take in unassociated elements of the three tiers and return an associated autosegmental representation. We first define the domain function $\phi_{\texttt{domain}}$. Because we want our transductions to occur for all binyanim, we set $\phi_{\texttt{domain}} \stackrel{\text{def}}{=} \texttt{True}$. Because the size of the output structure will be the same as the size of the input structure, it suffices to set the domain of the output structure $C \stackrel{\text{def}}{=} \{1\}$ and $\phi_{\texttt{license}} \stackrel{\text{def}}{=} \texttt{True}$; this simply indicates that the domain of the output structure is the same as that of the input structure. Next, we define for each relation in $\mathfrak{R}$ the transduction for this relation. Precedence is preserved in our output structure, so we have $\phi_<(x,y) = x < y$. For all relations in $\mathcal{P}$, $\mathcal{C}$, $\mathcal{R}$ and $\mathcal{M}$, these relations hold in the output only if they hold in the input, so for example $\phi_\texttt{m}(x) = m(x)$ for all $m \in \mathcal{M}$; the transduction

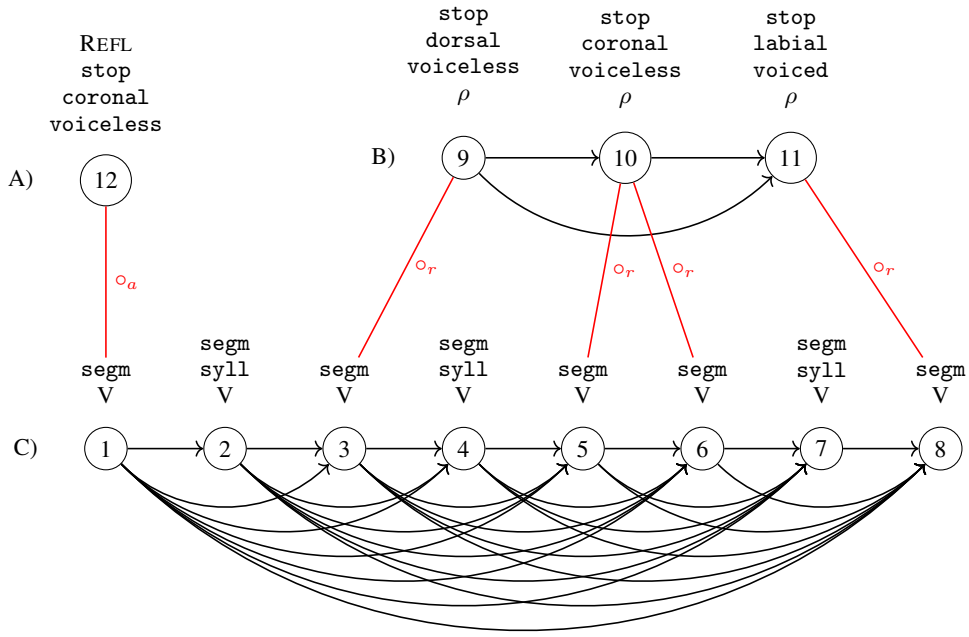


Figure 3: Representations of the affix [t-] (A), root *KTB* (B), CVCVCCVC template (C) and the associations (red) output by our transduction for *taKaTTaB* in binyan V.

formulae are analogous for the other three sets of unary relations. We now turn to the main contribution of the paper: the transductions for affixation $\phi_{\circ_a}$ (§3.2.1) and root association $\phi_{\circ_r}$ (§3.2.2). An example of the transduced associations for *taKaTTaB* in binyan V is given in Figure 3 with the associations highlighted in red; we will return to this example in §3.3.

3.2.1 Affixation: $\phi_{\circ_a}$

For the current purposes, we focus on the association between a single affix and its position in the prosodic template, but it is straightforward to see how this approach would extend to multiple affixes. To determine the appropriate slot(s) in the template, we define a predicate for the ordering relation over consonantal slots using the successor relation:

$$x \lhd_c y \stackrel{\text{def}}{=} \texttt{c_slot}(x) \wedge \texttt{c_slot}(y) \wedge x < y \wedge \nexists(z)[x < z < y \wedge \texttt{c_slot}(z)] \quad (17)$$

We then use this predicate to define the relevant positions of consonants in the prosodic template:

$$\texttt{first_c}(x) \stackrel{\text{def}}{=} \texttt{c_slot}(x) \wedge \nexists(y)[y \lhd_c x] \quad (18)$$

$$\texttt{last_c}(x) \stackrel{\text{def}}{=} \texttt{c_slot}(x) \wedge \nexists(y)[x \lhd_c y] \quad (19)$$

$$\texttt{second_c}(x) \stackrel{\text{def}}{=} \exists(y)[y \lhd_c x \wedge \texttt{first_c}(y)] \quad (20)$$

$$\texttt{third_c}(x) \stackrel{\text{def}}{=} \exists(y)[y \lhd_c x \wedge \texttt{second_c}(y)] \quad (21)$$

$$\texttt{fourth_c}(x) \stackrel{\text{def}}{=} \exists(y)[y \lhd_c x \wedge \texttt{third_c}(y)] \quad (22)$$

Similarly, we define predicates for the relevant positions in the affix:

$$\texttt{first_a}(x) \stackrel{\text{def}}{=} \texttt{affix}(x) \wedge (\nexists z)[\texttt{affix}(z) \wedge z < x] \quad (23)$$

$$\texttt{last_a}(x) \stackrel{\text{def}}{=} \texttt{affix}(x) \wedge (\nexists z)[\texttt{affix}(z) \wedge x < z] \quad (24)$$

It is these predicates, which make explicit the position of each of the finitely many consonantal slots and affixal elements, which allow us to define our affixation transduction using only First Order logic. For binyanim IV, V, VI, and VII, the prefix simply fills the first C slot of the prosodic template:

$$\texttt{simple_prefix}(x, y) \stackrel{\text{def}}{=} \texttt{first_c}(y) \wedge (\text{IV}(y) \vee \text{V}(y) \vee \text{VI}(y) \vee \text{VII}(y)) \quad (25)$$

For binyan VIII, I differ from McCarthy in that I associate the affix directly with the second C slot rather than with the first and then flipping it. As discussed in §3.1, however, either approach requires encoding information about the binyan in our representation, and our approach does not require ordered transductions, while a more direct implementation of McCarthy's flop rule would.

$$\texttt{VIII_infix}(x,y) \stackrel{\text{def}}{=} \text{VIII}(y) \wedge \texttt{second_c}(y) \tag{26}$$

Next, the [st-] prefix in binyan X associates the first consonant of the prefix to the first C slot of the prosodic template and the second to the second:

$$\texttt{X_prefix}(x,y) \stackrel{\text{def}}{=} \text{X}(y) \wedge ((\texttt{first_a}(x) \wedge \texttt{first_c}(y)) \vee (\texttt{last_a}(x) \wedge \texttt{second_c}(y))) \tag{27}$$

The infix in binyanim XII and XIV associates with the third C slot in the prosodic template:

$$\texttt{simple_infix}(x,y) \stackrel{\text{def}}{=} \texttt{third_c}(y) \wedge (\text{XII}(y) \vee \text{XIV}(y)) \tag{28}$$

The infix in binyan XIII associates with the third and fourth C slots. Following McCarthy, I assume that this infix consists of a single [w] on the morphological tier, which is associated with two positions in the prosodic template:

$$\texttt{XIII_infix} \stackrel{\text{def}}{=} \text{XIII}(y) \wedge (\texttt{third_c}(y) \vee \texttt{fourth_c}(y)) \tag{29}$$

Having defined our affix associations for all binyanim which take affixes, we define our overall affixation transduction as the disjunction of these: an association between elements x and y is a valid affix association if x is an element of an affix and one of the predicates defined above is satisfied:

$$\begin{aligned}\phi_{\circ_a}(x,y) \stackrel{\text{def}}{=} \; & \texttt{affix}(x) \\ & \wedge (\texttt{simple_prefix}(x,y) \\ & \vee \texttt{VIII_infix}(x,y) \\ & \vee \texttt{X_prefix}(x,y) \\ & \vee \texttt{simple_infix}(x,y) \\ & \vee \texttt{XIII_infix}(x,y))\end{aligned} \tag{30}$$

3.2.2 Root Association: $\phi_{\circ_r}$

We know from McCarthy (1981) that the root consonants will fill the remaining C slots in the prosodic template from left-to-right in general (albeit with binyanim II, V, and XIII violating this). We thus wish to determine whether a given C slot is still available for association after affixation has applied. There are two options for doing so. First, we can follow McCarthy and define it in terms of whether affixation has already applied, as follows:

$$\texttt{unassociated}(x) \stackrel{\text{def}}{=} \texttt{c_slot}(x) \wedge \nexists(y)[y \circ_a x] \tag{31}$$

This definition is in line with traditional autosegmental and generative rule-ordering approaches, and could be employed so long as $\phi_{\circ_r}$ is ordered *after* $\phi_{\circ_a}$, so that it can make reference to $\circ_a$. Because we encode information about the binyanim in our representations, however, we can instead define our predicate in terms of the C slots that we know *will* be associated with an affix. This approach allows $\phi_{\circ_r}$ and $\phi_{\circ_a}$ to remain unordered; i.e. they may apply simultaneously. The corresponding predicate is given below.

$$\begin{aligned}\texttt{unassociated}(x) \stackrel{\text{def}}{=} \; & \texttt{c_slot}(x) \\ & \wedge ((\neg \texttt{first_c}(x) \wedge \\ & (\text{IV}(x) \vee \text{V}(x) \\ & \vee \text{VI}(x) \vee \text{VII}(x))) \\ & \vee (\text{VIII}(x) \wedge \neg \texttt{second_c}(x)) \\ & \vee (\text{X}(x) \wedge \neg(\texttt{first_c}(x) \\ & \vee \texttt{second_c}(x))) \\ & \vee (\neg \texttt{third_c}(x) \\ & \wedge (\text{XII}(x) \vee \text{XIV}(x))) \\ & \vee (\text{XIII}(x) \wedge \neg(\texttt{third_c}(x) \\ & \vee \texttt{fourth_c}(x))))\end{aligned} \tag{32}$$

Each portion of the disjunction is straightforwardly derivable from the affixation predicates in §3.2.1; though somewhat redundant, this predicate allows us not to concern ourselves with ordering of our logical transductions. We now define an ordering over unassociated C slots in the prosodic template.

$$x \lhd_{\text{UC}} y \stackrel{\text{def}}{=} \texttt{unassociated}(x) \wedge \texttt{unassociated}(y) \wedge x < y \wedge \nexists(z)[x < z < y \wedge \texttt{unassociated}(z)] \tag{33}$$

We use this to define predicates for the positions of the unassociated consonants:

$$\texttt{first_UC}\,(x) \stackrel{\text{def}}{=} \texttt{unassociated}\,(x) \quad (34)$$
$$\wedge \nexists(y)[y \lhd_{\text{UC}} x]$$

$$\texttt{last_UC}\,(x) \stackrel{\text{def}}{=} \texttt{unassociated}\,(x) \quad (35)$$
$$\wedge \nexists(y)[x \lhd_{\text{UC}} y]$$

$$\texttt{second_UC}\,(x) \stackrel{\text{def}}{=} \exists(y)[y \lhd_{\text{UC}} x \quad (36)$$
$$\wedge \texttt{first_c}\,(y)]$$

$$\texttt{third_UC}\,(x) \stackrel{\text{def}}{=} \exists(y)[y \lhd_{\text{UC}} x \quad (37)$$
$$\wedge \texttt{second_c}\,(y)]$$

$$\texttt{fourth_UC}\,(x) \stackrel{\text{def}}{=} \exists(y)[y \lhd_{\text{UC}} x \quad (38)$$
$$\wedge \texttt{third_c}\,(y)]$$

We similarly define predicates for the position of the root consonants:

$$\texttt{first_r}\,(x) \stackrel{\text{def}}{=} \texttt{root}\,(x) \quad (39)$$
$$\wedge (\nexists z)[\texttt{root}\,(z) \wedge z < x]$$

$$\texttt{second_r}\,(x) \stackrel{\text{def}}{=} \texttt{root}\,(x) \quad (40)$$
$$\wedge (\exists z)[\texttt{root}\,(z) \wedge z < x]$$
$$\wedge (\exists z)[\texttt{root}\,(w) \wedge x < w]$$

$$\texttt{third_r}\,(x) \stackrel{\text{def}}{=} \texttt{root}\,(x) \quad (41)$$
$$\wedge (\nexists z)[\texttt{root}\,(z) \wedge x < z]$$

Again, it is these predicates, which make explicit the position of each of the finitely many elements of the prosodic template and consonantal root, which allow us to define the root association transduction using only First Order logic. The first consonant of the root *always* associates with the first available C slot, and the second *always* associates with the second. If there is a fourth unassociated C slot, it will always be associated to the final consonant of the root:

$$\texttt{assoc}\,(x,y) \stackrel{\text{def}}{=} (\texttt{first_UC}\,(y) \quad (42)$$
$$\wedge \texttt{first_r}\,(x))$$
$$\vee (\texttt{second_UC}\,(y) \wedge \texttt{second_r}\,(x))$$
$$\vee (\texttt{fourth_UC}\,(y) \wedge \texttt{third_r}\,(x))$$

The third consonant of the root will associate with the third unassociated C slot for all binyanim for which McCarthy's erasure rule does not apply, and the second one otherwise:

$$\texttt{erasure}\,(y) \stackrel{\text{def}}{=} \text{II}(y) \vee \text{V}(y) \vee \text{XII}(y) \quad (43)$$

$$\texttt{third}\,(x,y) \stackrel{\text{def}}{=} \texttt{third_UC}\,(y) \quad (44)$$
$$\wedge ((\texttt{erasure}\,(y) \wedge \texttt{second_r}\,(x))$$
$$\vee (\neg\,\texttt{erasure}\,(y) \wedge \texttt{third_r}\,(x)))$$

Our complete root-pattern association transduction is thus simply given by:

$$\phi_{\circ_{\text{r}}}(x,y) \stackrel{\text{def}}{=} \texttt{assoc}\,(x,y) \vee \texttt{third}\,(x,y) \quad (45)$$

3.3 Example

To illustrate the logical transductions defined in the previous section, we consider the derivation of *taKaTTaB* in binyan V. The representations of the prosodic CVCVCCVC template, root *KTB*, and affix [t-] are given in Figure 3.

Following §3.2, the domain of the output structure is the same as the input structure, and all features in $\mathcal{M}, \mathcal{P}, \mathcal{C}$, and $\mathcal{R}$ are preserved, as is the precedence relation $<$; the domain thus remains unchanged. We now turn to our First Order transductions $\phi_{\circ_{\text{a}}}$ and $\phi_{\circ_{\text{r}}}$. We consider the former first for ease of exposition, but recall from §3.2 that these transductions can be — and are in the current analysis — applied simultaneously.

We know that $\texttt{affix}\,(12) = \texttt{True}$, where 12 refers to domain element 12 (Figure 3 A). We also know that for all elements y in our prosodic template (Figure 3 C), we have $\text{V}(y) = \texttt{True}$. As such, the only predicate in the disjunction of $\phi_{\circ_{\text{a}}}$ that we might satisfy is $\texttt{simple_prefix}$, since all other predicates require a different binyan. From Equation 25, we know that we must also have $\texttt{first_c}\,(y)$, which is true of element 1; we thus have $12 \circ_a 1$. No other associations are licensed by $\phi_{\circ_{\text{a}}}$ so this transduction is complete.

Next, we consider our root association relation $\phi_{\circ_{\text{r}}}$. From Equation 32, we have that $\texttt{unassociated}\,(y) = \texttt{True}$ for $y \in \{3, 5, 6, 8\}$. We begin by considering the general associations given by $\texttt{assoc}$ (Equation 42). We have that $\texttt{first_UC}\,(3) = \texttt{True}$ and $\texttt{first_r}\,(9) = \texttt{True}$, and thus $9 \circ_r$

3. Similarly, $\texttt{second_UC}(5) = \texttt{True}$ and $\texttt{second_r}(10) = \texttt{True}$, so we have $10 \circ_r 5$. Finally, we have $\texttt{fourth_UC}(8) = \texttt{True}$ and $\texttt{third_r}(11) = \texttt{True}$, so $11 \circ_r 8$. This exhausts `assoc`, and we turn to `third`. We have $\texttt{erasure}(y) = \texttt{True}$ for all y that are domain elements of the template (Figure 3 C), and $\texttt{third_UC}(6) = \texttt{True}$ and $\texttt{second_r}(10) = \texttt{True}$, so by Equation 44, $10 \circ_r 6$, concluding our transduction for $\phi_{\circ_{\texttt{r}}}$. The output structure is visualized in Figure 3, with the outputs of the logical transductions $\phi_{\circ_{\texttt{r}}}$ and $\phi_{\circ_{\texttt{a}}}$ highlighted in red.

4 Discussion

The transductions defined in this paper have served to demonstrate that the constraints of the Arabic system — in particular, the finite length of the consonantal root — make it possible to define transductions corresponding to the autosegmental analysis of McCarthy (1981) using only First Order logic, despite such transductions not being MSO-definable for autosegmental representations of arbitrary length (Jardine, 2017). I highlight two directions for future work.

Firstly, while a majority of roots in Arabic are triconsonantal, the length of the consonantal root varies from 2 to 5, and McCarthy extends his analysis to encompass those with 4. Though this analysis focuses on Modern Standard Arabic, the specifics of the association transductions vary across Arabic dialects, and future work should extend the approach here to encompass this variation. This analysis could also be extended to other Semitic languages, including Hebrew, which allows for even greater complexity in the consonantal root. Importantly, what allows for the present result is the *finite* number of consonants in the root, rather than a *specific* exact number of these consonants. It is thus expected that the association transductions would be straightforwardly First Order definable for roots consisting of some other finite number of consonants, though these transductions may be largely separate from those used for the triconsonantal roots here. Indeed, in extending the present analysis, it will be important to weigh tradeoffs between capturing apparent generalizations and remaining at least MSO-definable.

Secondly, the approach presented here focuses on the logical complexity of autosegmental approaches to root-and-pattern morphology, but a great deal of other approaches exist in the literature (Hudson, 1986; Ussishkin, 2003; Arad, 2005; Tucker, 2011; Alqarni, 2022, i.a.). Several of these approaches involve defining a well-formed output in an Optimality-Theoretic sense rather than defining a *transduction* between a set of unassociated input elements and the associated output. Given the limits of MSO-definability for autosegmental approaches in the general case (Jardine, 2017), and the insights gained by the transductions defined in this paper, it would be interesting to take a parallel approach of logical formalization to these other frameworks. Such an investigation may yield new insight into existing theoretical debates by highlighting which properties of the Arabic system allow for different theories to be computationally tractable: for autosegmental phonology, it is the finite nature of the root, but for Optimility-Theoretic approaches, it may be something else entirely. Considering the different properties on which the tractability of these theories hinges may give new insight into which approaches should be favored from a cognitive perspective.

5 Conclusion

In this paper, I have demonstrated that the logical transductions corresponding to McCarthy's three-tier autosegmental analysis of Arabic derivational morphology are definable with First Order logic, and I introduced the corresponding relational structures and association transductions. This First Order definability is due to the small, finite length of the consonantal root in Arabic, and is in contrast to cases of arbitrary length, where left-to-right autosegmental association is not even MSO-definable (Jardine, 2017). Indeed, the relatively constrained set of possible lengths for Semitic consonantal roots is an interesting property of these languages, and the analysis presented here demonstrates a computational advantage of this property of the system.

References

Muteb Alqarni. 2022. Word-formation and reduplication in standard Arabic: A new distributed morphology approach. *Poznan Studies in Contemporary Linguistics*, 58(3):381–418.

Maya Arad. 2005. *Roots and patterns: Hebrew morpho-syntax*. Springer.

J Richakd Büchi. 1960. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, pages 66–92.

Noam Chomsky. 1957. *Syntactic structures*. Walter de Gruyter.

Noam Chomsky. 1963. Formal properties of grammars. *Handbook of math. Psychology*, 2:328–418.

George N Clements and Kevin C Ford. 1979. Kikuyu tone shift and its synchronic consequences. *Linguistic inquiry*, pages 179–210.

GN Clements and SJ Keyser. 1983. From CV phonology: A generative theory of the syllable.

John Goldsmith. 1976. *Autosegmental phonology*. Ph.D. thesis, Doctoral dissertation/Garland Press.

Thomas Graf. 2022. Subregular linguistics: Bridging theoretical linguistics and formal grammar. *Theoretical Linguistics*, 48(3-4):145–184.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. *Phonological typology, phonetics and phonology*, 23:126–195.

Grover Hudson. 1986. Arabic root and pattern morphology without tiers. *Journal of Linguistics*, 22(1):85–122.

Adam Jardine. 2017. On the logical complexity of autosegmental representations. In *Proceedings of the 15th Meeting on the Mathematics of Language*, pages 22–35, London, UK. Association for Computational Linguistics.

Adam Jardine. 2024. Autosegmental Representations in Zigula and Shambaa. In *Doing Computational Phonology*.

Ronald M Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational linguistics*, 20(3):331–378.

John J McCarthy. 1981. A prosodic theory of nonconcatenative morphology. *Linguistic inquiry*, 12(3):373–418.

Matthew A Tucker. 2011. The morphosyntax of the Arabic verb: Toward a unified syntax-prosody.

Adam Ussishkin. 2003. Templatic effects as fixed prosody: The verbal system in Semitic. *Amsterdam Studies in the Theory and History of Linguistic Science Series 4*, pages 511–530.

Comonadic Morphophonology: A Compositional Framework for Context-Dependent Morphological Rules in Finnish

Yongseok Jang[*]
Independent Researcher
yongsk0066@gmail.com

Abstract

Composing finite-state transducers (FSTs) for context-dependent morphophonological rules—consonant gradation, vowel harmony, possessive suffix assimilation—leads to multiplicative state explosion; neural models sidestep the problem but provide no formal account of the rules themselves. We present the first framework where each morphophonological rule is a function from a focused local context to a single output segment—the type of a local rule familiar from cellular automata—and where length-changing rules compose as coKleisli arrows of a comonad. Our central contribution is the *Writer comonad* (DeletionSet $\times$ Zipper), a new algebraic construction that restores strict coKleisli compositionality for such rules: each rule is a coKleisli arrow, extend lifts it to a global transformation, and deletions accumulate as a monoid action rather than requiring intermediate materialization. As supporting evidence, thirteen coKleisli arrows provide an alternative formulation expressing the same morphophonological behaviors that Omorfi encodes via 874 continuation classes (67:1 reduction at the rule-representation level), and the same abstraction enables bidirectional morphology—a MorphGenerator reuses the analysis arrows for generation. On UD Finnish-TDT, the system achieves 83.92% UPOS accuracy with rule-only disambiguation (94.66% with an external suffix tagger), validating the framework as a practical morphological engine.

1 Introduction

A standard FST treats a rule as a string-to-string relation. We instead treat each rule as a function from a focused local context to a single output segment—precisely the type of a local rule in cellular automata. Composing such arrows is not function composition: the output of one rule has type Σ, not Zipper$\langle\Sigma\rangle$. The coKleisli construction over a Zipper comonad fills exactly this typing gap.

We show that morphophonological rules—including length-changing operations such as consonant deletion—are coKleisli arrows of a Writer comonad, yielding the first framework that handles length-changing morphophonological rules within a compositional category-theoretic structure. Thirteen coKleisli arrows provide an alternative formulation expressing the same morphophonological behaviors that Omorfi encodes via 874 continuation classes (67:1 reduction at the rule-representation level via orthogonal composition), and the same abstraction unifies character-level, morpheme-level, and sentence-level processing.

Finnish morphophonology motivates the framework. *Consonant gradation* alternates stem-final consonants between strong and weak grades depending on syllable structure; *vowel harmony* requires suffix vowels to agree in backness with stem vowels; and *possessive suffix vowel copying* inserts a vowel identical to the preceding segment at morphological boundaries. Rule-based systems such as Voikko (Pitkänen, 2006) and Omorfi (Pirinen, 2015) encode these alternations as FST cascades (Koskenniemi, 1983; Beesley and Karttunen, 2003), achieving excellent coverage but limited compositionality: composing two FSTs yields a product automaton whose state space grows multiplicatively. Neural approaches (Kanerva et al., 2018; Nguyen et al., 2021) achieve high accuracy but provide no formal account of the underlying processes. No existing framework treats morphophonological rules as first-class compositional objects with formal combination laws.

Our goal is not to re-derive known rules, but to provide a *composition algebra* for rules that FST cascades handle only through state-space multiplication. The Writer comonad—which tracks deletions as a monoid action, deferring materialization to pipeline end—is a new algebraic construction

[*]Claude (Anthropic) was used for LaTeX polishing and code-writing assistance; the author takes full responsibility for the content.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 100–112
July 3-4, 2026 ©2026 Association for Computational Linguistics

with no prior analogue in computational morphology. In contrast to FST ε-transitions, which require recomposition after each length-changing rule, the Writer comonad accumulates deletions within the coKleisli pipeline, preserving strict associativity without intermediate re-materialization.

Morphophonological rules are naturally *coKleisli arrows*: functions $W\,a \to b$ where W is a comonad providing a context window around a focused element. The comonad operation extend lifts such a local rule into a global transformation, and coKleisli composition ($\gg\!=\!\gg$) chains rules with associativity and identity guarantees. This connection is not accidental: Capobianco and Uustalu (2010) established that cellular automaton local behavior is precisely a coKleisli map of the exponent comonad, and morphophonological rule application instantiates exactly this pattern.

Our primary contribution is the Writer comonad (DeletionSet $\times$ Zipper), which restores strict coKleisli compositionality for length-changing morphophonological rules—a problem that FST cascades leave unresolved without multiplicative state explosion (§3, §3.6). As supporting demonstrations:

- The same comonadic abstraction operates at multiple linguistic levels—character (gradation), morpheme (harmony), and sentence (Constraint Grammar (CG) lite disambiguation)—via the uniform Zipper/extend mechanism (§3.5).
- The coKleisli pipeline enables bidirectional morphology: a MorphGenerator reuses the same arrows for both analysis and generation (§3.4).

We evaluate on UD Finnish-TDT, reporting 83.92% UPOS with rule-only disambiguation (94.66% with a suffix tagger), per-rule latency microbenchmarks, and a 67:1 reduction at the rule-representation level (§5).

2 Background

2.1 Finnish Morphophonology

Consonant gradation (*astevaihtelu*) alternates stem-final consonants between a *strong grade* and a *weak grade* (Hakulinen et al., 2004). The grade is determined by syllable structure: open syllables trigger the strong grade, closed syllables the weak grade. The alternation affects stops and certain consonant clusters according to the patterns in Table 1.

#	Strong	Weak	Type	Example
1	pp	p	Quant.	kaappi → kaapi
2	tt	t	Quant.	matto → mato
3	kk	k	Quant.	kukka → kuka
4	p	v	Qual. (s)	tupa → tuva
5	t	d	Qual. (s)	katu → kadu
6	k	∅	Qual. (s)	puku → puu
7	mp	mm	Qual. (c)	kampa → kamma
8	lt	ll	Qual. (c)	kulta → kulla
9	nt	nn	Qual. (c)	ranta → ranna
10	rt	rr	Qual. (c)	parta → parra
11	nk	ng	Qual. (c)	kenkä → kengä

Table 1: The 11 Finnish consonant gradation patterns (KOTUS types). Quant. = quantitative (geminate reduction), Qual. = qualitative (s = single, c = cluster). CoKleisli arrows examine only the immediate left neighbor, consistent with input strictly local (ISL) functions (Chandlee, 2014).

Vowel harmony (*vokaalisointu*) divides Finnish vowels into back (*a, o, u*), front (*ä, ö, y*), and neutral (*e, i*). Suffixes with archiphonemes A, O, U are realized as back or front variants depending on the stem. Vowel harmony exhibits TSL-2 locality (Heinz, 2018): the harmony arrow scans leftward through transparent neutral vowels, depending only on the nearest non-neutral vowel on the vowel tier (a window of size 2 on that tier). The Zipper's full left context naturally supports this tier projection: the harmony arrow scans leftward past neutral vowels until a non-neutral vowel is found, implementing this tier-based locality within the standard coKleisli mechanism.

Possessive suffix vowel copying. The archiphoneme *V* is realized as a copy of the immediately preceding vowel: the third-person possessive suffix *-Vn* becomes *-an* after *a*, *-en* after *e*, and analogously for the remaining vowels.

2.2 Comonads in Computer Science

A *comonad* is dual to a monad: where monads abstract over computations that produce effects, comonads abstract over computations that consume context (Orchard and Mycroft, 2012). We use the extract/extend formulation:

Definition 1 (Comonad)**.** *A comonad* $(W, \mathsf{extract}, \mathsf{extend})$ *consists of an endofunctor* W *with* $\mathsf{extract} : W\,a \to a$ *and* $\mathsf{extend} : (W\,a \to b) \to W\,a \to W\,b$*, satisfying:*

$$\mathsf{extend}\ \mathsf{extract} = \mathrm{id} \tag{L1′}$$

$$\mathsf{extract} \circ \mathsf{extend}\ f = f \tag{L2′}$$

$$\mathsf{extend}\ g \circ \mathsf{extend}\ f = \mathsf{extend}\ (g \circ \mathsf{extend}\ f) \tag{L3′}$$

Definition 2 (CoKleisli arrow)**.** *A function* $f : W\ a \to b$ *is a* coKleisli arrow. *CoKleisli arrows compose via* $(f \gg\!= g)(w) = g(\mathsf{extend}\ f\ w)$*, forming the* coKleisli category *of* W *(associative, with* extract *as identity).*

The ListZipper comonad. The ListZipper (Huet, 1997) represents a non-empty sequence with a distinguished *focus* and left/right context (Figure 1). extract returns the focused element; extend f applies f at every position, each seeing the full context from that position's perspective. Uustalu and Vene (2005) showed that this captures the essence of applying a local update rule globally—precisely a cellular automaton. Our insight is that Finnish morphophonological rules instantiate the same pattern.

3 Formalization

3.1 Consonant Gradation as CoKleisli Arrows

We encode each consonant gradation pattern as a two-character window (c_0, c_1), where c_0 is the left context and c_1 the focused character. The coKleisli arrow has the signature:

$$\mathsf{apply_gradation} : \mathsf{Zipper}(\mathit{char}) \times \mathit{Grade} \to \mathit{char} \tag{1}$$

where $\mathit{Grade} \in \{\mathsf{Strong}, \mathsf{Weak}\}$. Partially applying the grade yields a coKleisli arrow $\mathsf{Zipper}(\mathit{char}) \to \mathit{char}$ suitable for extend.

Design principle: only position 1 is transformed. The coKleisli arrow modifies only the focused character (position 1 of the pattern window) and treats the left neighbor (position 0) as read-only context. This is essential because extend applies the function at *every* position: if both characters were eligible, geminate $pp \to p$ would produce double-counting. By transforming only position 1, each position produces exactly one output, consistent with comonadic extend semantics. This mirrors cellular automata, where each cell reads neighbors but writes only its own state.

The deletion marker. Pattern 6 (intervocalic *k* deletion) presents a type-theoretic challenge: the coKleisli arrow has type $\mathsf{Zipper}(\mathit{char}) \to \mathit{char}$, requiring each position to produce exactly one character. A naïve approach inserts a null character `'\0'` as a deletion marker, but this breaks strict coKleisli associativity because filtering between steps requires materializing and re-creating zippers. We resolve this algebraically via the Writer comonad (§3.6).

Pattern priority and suppression. Priority is determined by specificity: geminates match first, then clusters, then single consonants. A suppression predicate $\mathsf{is_pos0}(c, r)$ returns true when the focused character c and its right neighbor r form position 0 of a higher-priority pattern (e.g., $\mathsf{is_pos0}(\mathtt{p}, \mathtt{p})$ = true for the geminate pp), preventing single-consonant rules from firing erroneously. Formally:

$$\mathsf{grad}(z, W) = \begin{cases} c & \mathsf{is_pos0}(c, r) \\ \mathsf{tgt}(p, W)[1] & \mathsf{find}(l, c, r, W) = \mathsf{Some}(p) \\ c & \text{otherwise} \end{cases} \tag{2}$$

3.2 Worked Example: Weakening "kaappi"

We trace $\mathsf{extend}(\mathsf{apply_gradation}(\cdot, \mathsf{Weak}))$ on *kaappi* in Figure 3. The key interactions are: (1) at position 3, the first p is *suppressed* because $\mathsf{is_pos0}(\mathtt{p}, \mathtt{p})$ recognizes it as position 0 of the geminate pattern—without suppression, the single-consonant rule would fire, yielding incorrect **kaavi*; (2) at position 4, the geminate pattern matches and the Writer comonad arrow returns $(\{4\}, \mathtt{p})$, marking the position for deletion; (3) materialize applies the accumulated $\mathit{DeletionSet} = \{4\}$, yielding *kaapi* (weak grade).

3.3 Vowel Harmony as a CoKleisli Arrow

Vowel harmony is a coKleisli arrow $\mathsf{harmony} : \mathsf{Zipper}(\mathit{char}) \to \mathit{char}$ resolving archiphonemes by harmony class:

$$\mathsf{detect}(z) = \begin{cases} \mathsf{Back} & \text{back vowel left} \\ \mathsf{Front} & \text{front vowel left} \\ \mathsf{Front} & \text{neutral only} \end{cases} \tag{3}$$

$$\mathsf{harmony}(z) = \begin{cases} a/\ddot{a} & \mathsf{extract}(z) = \mathtt{A} \\ o/\ddot{o} & \mathsf{extract}(z) = \mathtt{O} \\ u/y & \mathsf{extract}(z) = \mathtt{U} \\ c & \text{otherwise} \end{cases} \tag{4}$$

Example 1. *talo* + *-ssA* → t-a-l-o-s-s-a (back vowel *a* found left). Conversely, *kynä* + *-ssA* → k-y-n-ä-s-s-ä (front vowels *y, ä*).

3.4 Pipeline Composition and Bidirectional Morphology

The morphophonological pipeline chains three coKleisli arrows (Figure 2):

$$\mathsf{pipe} = \mathsf{grad} \gg\!= \mathsf{harmony} \gg\!= \mathsf{poss} \tag{5}$$

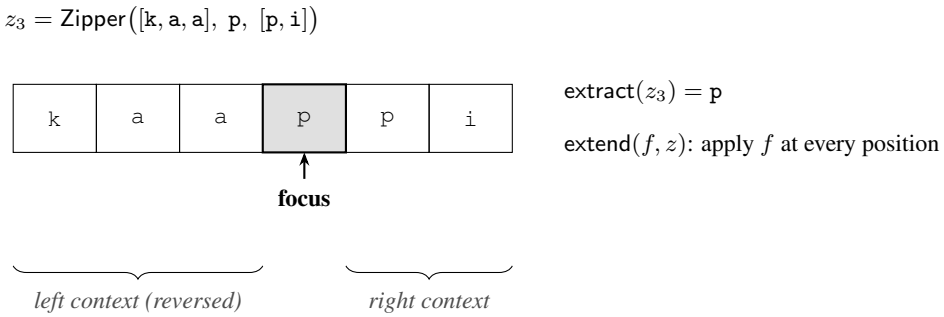


Figure 1: The ListZipper data structure for the word *kaappi* focused on position 3. The left context is stored in reversed order (nearest neighbor last) for $O(1)$ access. extract returns the focused element; $\mathsf{extend}(f, z)$ applies a coKleisli arrow f at every position, each seeing the full context from that position's perspective.

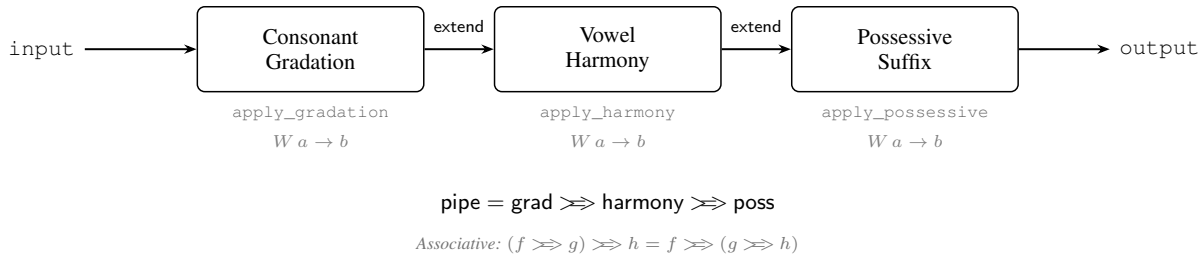


Figure 2: Morphophonological pipeline as coKleisli composition. Each box is a coKleisli arrow ($W\ a \to b$) lifted globally by extend. CoKleisli composition is associative with extract as identity.

A naïve implementation inserts a `'\0'`-filtering step between gradation and harmony, which breaks strict associativity. The Writer comonad (§3.6) resolves this: each arrow returns $(DeletionSet, char)$, accumulated deletions are merged via monoid union during extend, and a single materialize step applies all deletions at pipeline end. This restores strict coKleisli associativity across the full pipeline, including deletion.

Example 2 (Pipeline: "kampAstAVn" (weak grade)). (1) Gradation: `mp` → `mm` ⇒ `kammAstAVn`. (2) Harmony: `A` → `a` ⇒ `kammastaVn`. (3) Possessive: `V` → `a` ⇒ `kammastaan` ('from his/her comb').

MorphGenerator: bidirectional morphology. The coKleisli pipeline enables bidirectional morphology: analysis (via Voikko FST, hereafter VFST) and generation (via coKleisli arrows) share the same rules. The MorphGenerator produces inflected forms from baseform + features by applying coKleisli arrows in the generative direction (currently 11 noun cases). Since each rule is an explicit function rather than a compiled FST state, the same logic drives both analysis and generation with minimal additional code.

Rule ordering. Gradation precedes harmony because gradation may delete characters, changing the vowel context. The coKleisli framework does not eliminate order-dependence—it formalizes it: each rule remains individually addressable and testable, and the composition order is explicit rather than implicit in merged automaton states.

3.5 CG Rules as Comonadic Extension

The comonadic framework extends to sentence-level disambiguation: only the Zipper's element type changes (from character to ReadingSet), while the coKleisli composition structure is identical (the deletion-handling Writer comonad needed for character-level rules is formalized in §3.6). In CG-lite, each rule is a coKleisli arrow over *reading sets*:

$$rule : \mathsf{Zipper}(ReadingSet) \to ReadingSet \quad (6)$$

Twenty-four rule types are implemented, covering removal, selection, and structural patterns parameterized by part-of-speech (POS) tags, baseforms, and feature constraints. The current Finnish configuration contains 62 active rules (85 total; 23 disabled for accuracy after tuning). All rules enforce a safety invariant: a rule never removes the last reading at any position, ensuring well-definedness

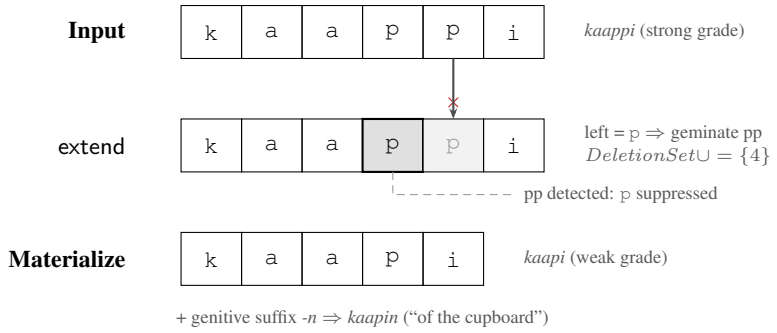


Figure 3: Step-by-step trace of consonant gradation on *kaappi* using the Writer comonad (positions are 0-indexed). The extend pass applies the gradation arrow at each position. At position 3, the first p is recognized as position 0 of the geminate pp pattern and **suppressed** (not transformed to v). At position 4, the second p matches position 1 of the geminate pattern; the arrow returns $(\{4\}, \mathrm{p})$, marking it for deletion via the $DeletionSet$ while preserving the character in the zipper. Materialize applies accumulated deletions once, yielding *kaapi* (weak grade stem).

Level	Zipper	Arrow	Out
Char	$\mathsf{Zip}\langle c\rangle$	Gradation	c
Morph	$\mathsf{Zip}\langle c\rangle$	Harmony	c
Sent	$\mathsf{Zip}\langle RS\rangle$	CG rule	RS

Table 2: Unified comonadic abstraction across three linguistic levels (c = char, RS = ReadingSet). Zipper provides context, the coKleisli arrow computes a local transformation, extend lifts it globally. These three levels are operationalized in MCE at: char/morph in the MorphGenerator (§3.4), sent in CG-lite disambiguation (§3.5).

of subsequent extend passes. The same coKleisli abstraction that governs character-level morphophonology applies at this sentence level—each CG rule is a coKleisli arrow over reading sets, composed via extend (details in Appendix B.3).

Multiple CG rules are applied by iterating extend:

$$\mathsf{cg}(s, [r_1, \ldots, r_k]) = \mathsf{vec}(\mathsf{ext}(r_k, \ldots \mathsf{ext}(r_1, \mathsf{Zip}(s)) \ldots)) \quad (7)$$

Note that sequential extension is equivalent to coKleisli composition: $\mathsf{extend}(r_2) \circ \mathsf{extend}(r_1) = \mathsf{extend}(r_1 \gg\!= r_2)$, so the CG pipeline uses the same composition mechanism as the morphophonological pipeline ((5)).

Unified comonadic abstraction. The same algebraic structure operates at three linguistic levels (Table 2):

To our knowledge, this is the first application of the comonad/cellular-automaton correspondence (Capobianco and Uustalu, 2010) simultaneously at multiple linguistic levels in computational morphology.

3.6 Writer Comonad for Deletion

The deletion marker approach (§3.1) would break strict coKleisli associativity: filtering `'\0'` characters between extend passes requires materializing intermediate sequences and reconstructing zippers, violating the comonad law $\mathsf{extend}\ g \circ \mathsf{extend}\ f = \mathsf{extend}\ (g \circ \mathsf{extend}\ f)$.

The key idea: instead of immediately deleting characters, we *accumulate* deletion positions alongside the computation, applying all deletions once at pipeline end. We formalize this with the *Writer comonad*[1] $DS \times \mathsf{Zipper}$, where $(DS, \cup, \emptyset)$ is the *deletion monoid*—a set of positions with union as the monoid operation and the empty set as identity.

Definition 3 (Writer Comonad)**.** *Let $(DS, \cup, \emptyset)$ be the deletion monoid. The* Writer comonad *is the product $DS \times \mathsf{Zipper}$ with:*

$$\mathsf{extract}_W(w, z) = \mathsf{extract}(z) \quad (8)$$
$$\mathsf{extend}_W(f)(w, z) = \Big(w \cup \textstyle\bigcup_i \pi_1(f(w, z_i)), \mathsf{fmap}(\pi_2 \circ f)(w, z)\Big)$$

where z_i ranges over all positions of z, π_1/π_2 are the product projections onto DS and the output character, and fmap *denotes the underlying Zipper's* extend *applied to the character component: concretely,* $\mathsf{fmap}(\pi_2 \circ f)(w, z) = \mathsf{Zipper.extend}(\lambda z_i.\ \pi_2(f(w, z_i)))(z)$.

The construction factorizes: the Zipper satisfies standard comonad laws on characters, while DS

[1]The name "Writer comonad" is by analogy with the Writer monad—both accumulate a monoid value across a computation—but the construction is not the categorical dual of Writer. Standard categorical taxonomy does not provide a canonical name for this structure; we retain "Writer comonad" for its descriptive value with this caveat.

satisfies the monoid laws independently; extend_W preserves both.

Concretely, each coKleisli arrow returns (DS, A) instead of A: gradation returns $(\{pos\}, c)$ when deleting a consonant and $(\emptyset, c')$ otherwise; harmony and possessive always return $(\emptyset, c')$ since they are non-deleting. The pipeline begins with $w = \emptyset$ (no prior deletions); subsequent arrows in the coKleisli composition receive the accumulated DS from all preceding arrows via extend_W, propagating deletion information through the pipeline without intermediate materialization. A single materialize step at pipeline end applies all accumulated deletions.

Proposition 1. *The Writer comonad construction* $(DS \times \mathsf{Zipper},\ \mathsf{extract}_W,\ \mathsf{extend}_W)$ *satisfies the comonad laws on the character component and the monoid laws on DS, jointly ensuring compositional correctness of the pipeline.*

Proof sketch. **L1′** ($\mathsf{extend}_W\ \mathsf{extract}_W = \mathrm{id}$): For the deletion component, $\pi_1(\mathsf{extract}_W(w, z_i)) = \emptyset$ at every position, so $w \cup \bigcup_i \emptyset = w$ by monoid identity. For the character component, $\mathsf{Zipper.extend}(\lambda z_i.\ \pi_2(\mathsf{extract}_W(w, z_i)))(z) = \mathsf{Zipper.extend}(\mathsf{extract})(z) = z$ by the underlying Zipper's L1′. Thus $\mathsf{extend}_W(\mathsf{extract}_W)(w, z) = (w, z)$. **L2′** ($\mathsf{extract}_W \circ \mathsf{extend}_W\ f = \pi_2 \circ f$): $\mathsf{extract}_W(\mathsf{extend}_W(f)(w, z)) = \mathsf{extract}(\mathsf{Zipper.extend}(\lambda z_i.\ \pi_2(f(w, z_i)))(z)) = \pi_2(f(w, z))$ by the Zipper's L2′; the DS component is accumulated by extend_W and consumed by materialize. **L3′**: see Appendix A.6 for the full argument; the key step is that extend_W distributes over coKleisli composition because the DS components associate via the monoid law $(DS, \cup, \emptyset)$ and the character components associate via the underlying Zipper's L3′. All laws are verified by 44 unit tests covering the Zipper comonad laws (character component) and the DS monoid laws independently. □

The equivalence between the Writer pipeline and the old hybrid pipeline is verified on all 11 gradation patterns, confirming that the algebraic treatment of deletion introduces no behavioral difference.

4 Implementation

We implement the framework in the Morphological Computation Engine (MCE), a Rust (2024 edition, stable 1.86+) library targeting WebAssembly (∼380 KB). Since Rust lacks higher-kinded types, the comonad operations are implemented directly on the concrete `Zipper<T>` type rather than via a generic `Comonad` trait. Each morphophonological rule is a plain function `fn(&Zipper<char>) -> char`; CG rules use trait-based dispatch with the same coKleisli signature. The current extend allocates $O(n^2)$ due to cloning at each position, which is negligible for Finnish word lengths ($n \leq 15$). The implementation is verified through 317 tests (comonadic pipeline; 1,619 total across the system) covering Zipper comonad laws, *DeletionSet* monoid laws, all 11 gradation patterns in both directions, roundtrip properties, Writer pipeline equivalence, and CG rule correctness. Full implementation details, Rust code listings, and test methodology are provided in Appendix A.

5 Evaluation

5.1 Experimental Setup

We evaluate on UD Finnish-TDT v2.14 (Haverinen et al., 2014), distributed under CC-BY-SA 4.0, dev split (1,364 sentences, ∼25,000 tokens) with gold tokenization. Since our CG rules are hand-written and the suffix tagger was trained exclusively on the training split, the development set provides an unbiased evaluation. Accuracy is micro-averaged token-level accuracy; out-of-vocabulary (OOV) tokens (0.36%) default to NOUN. The morphological analyzer uses the Voikko Finnish morphological dictionary (voikko-fi v2.6, 39,607 lexical entries; `mor.vfst`, 4 MB). All experiments run on a single CPU core (Apple M2 Max); no GPU is required.

5.2 Morphophonological Correctness

All 11 consonant gradation patterns are correctly formalized as coKleisli arrows, verified by the Writer comonad pipeline. Of these, 7 exhibit full bidirectional roundtrip correctness ($\mathsf{strong}(\mathsf{weak}(w)) = w$); the 4 non-roundtrip cases all involve deletion, confirming deletion as the sole source of irreversibility. The Writer comonad laws hold as follows. L1′ (left identity) and L2′ (right identity) hold for arbitrary coKleisli arrows. L3′ (associativity) holds on the value (Zipper) component; the log component is treated as an accumulating output rather than a structural part of the comonad, and is materialized once at the end of each pipeline stage. In practice we invoke extend

Component	Mean (μs)	Std (μs)
CoKleisli (char-level)		
Gradation (avg/11)	0.61	0.24
Harmony (avg/4)	0.85	0.22
Possessive (avg/3)	0.70	0.24
Full pipeline (avg/4)	2.66	0.58
CG-lite (sentence-level)		
Single rule (avg/20)	7.66	1.34
Full CG (62 rules)	376.04	32.35

Table 3: Per-rule latency (10K iterations, Apple M-series). CoKleisli arrows: sub-μs; full pipeline: under 3 μs.

System	UPOS	Lemma	Approach
TNPP†	96.91	95.54	Neural
MCE-full (ours)	**94.66**	**93.09**	**FST+CG+stat**
Omorfi 1-best	83.88	82.63	FST + freq.
MCE-rules (ours)	83.92	93.09	FST+CG

Table 4: UPOS accuracy (%) on UD Finnish-TDT dev set with gold tokenization. Coverage: 99.35%. MCE-rules uses only comonadic CG disambiguation (62 active rules); MCE-full adds a lightweight suffix tagger (§5.4). †TNPP = Turku Neural Parser Pipeline. TNPP and Omorfi figures from Pirinen (2019), evaluated on an earlier UD Finnish-TDT version; POS annotation guidelines have remained stable across versions, so direct comparison is approximate but meaningful.

once per stage, so the question of log-component associativity under repeated extend does not arise in our pipeline; the Zipper alone satisfies all three comonad laws. Forty-four dedicated tests verify these properties. Full pattern-level results are in Appendix B.4.

5.3 System Performance

Throughput. The complete MCE pipeline processes **84,973 tokens/sec** on Apple M-series (∼12 μs/token, ∼0.21 ms for a 15-word sentence), demonstrating that coKleisli composition introduces no significant overhead compared to monolithic implementations.

Per-rule latency. Table 3 reports per-rule microbenchmarks (10K iterations). The full morphophonological pipeline averages 2.66 μs per word, confirming that coKleisli composition introduces negligible overhead beyond the sum of its components.

UPOS accuracy. With rule-only disambiguation (62 active CG rules), MCE achieves 83.92% UPOS and 93.09% lemma accuracy (Table 4) with 99.35% coverage. Adding a suffix tagger raises UPOS to 94.66%, narrowing the gap to TNPP from 12.99 pp to 2.25 pp.

Property	Comonadic	FST cascade
Composition	$O(k)$ functions	$O(\|S_1\| \times \|S_2\|)$
Modularity	Independent fns	Merged FST
Debugging	Per-rule tests	Composite trace
Deletion	Writer comonad	ε-transitions
Rule count	20 arrows‡	8,157 LEXICONs

Table 5: Comonadic vs. FST-based rule encoding. ‡20 total arrows: 13 morphophonological (11 gradation + harmony + possessive) + 7 system-level (tokenization, OOV fallback, case mapping, compound boundary, AUX resolution, coverage, Viterbi integration); at the morphophonological level, 13 vs. 874 (67:1).

Comparison with FST-based systems. Table 5 summarizes the key differences. Voikko's 8,157 LEXICON entries encode the full lexicon (stems, paradigms, compounds, *and* morphophonological rules), so the gross 408:1 ratio overstates the difference. At the morphophonological-rule-only level, Omorfi uses 874 continuation classes (810 harmony × suffix-class variants, 62 harmony × gradation, 2 gradation only)[2] vs. MCE's 13 arrows (67:1 at the rule-representation level). This comparison concerns rule-representation units: the MCE analysis pipeline itself uses a lexicon-based analyzer (Voikko-derived VFST) compatible with Omorfi-style approaches. The insight is *orthogonal composition*: FSTs require $O(P \times H \times C)$ entries per combination, while coKleisli composition is $O(P + H + C)$. The FST approach retains advantages within the same module: native ε-transitions, minimization, and direct encoding of lexical exceptions. Our framework operates in the SPE-era modular tradition (Chomsky and Halle, 1968), scoping itself to the rule layer and delegating lexical idiosyncrasies to the lexicon—in MCE, the Voikko-derived analyzer.

Developer ergonomics. Adding a new morphophonological rule requires defining a single coKleisli arrow (∼5 lines of Rust) that automatically composes with all existing rules. In contrast, adding a rule to an FST system requires modifying potentially hundreds of `LEXICON` entries to account for all interaction combinations—a direct consequence of the multiplicative vs. additive composition difference.

[2]Counted from Omorfi's `continuations.tsv` (commit `6ce67ac`).

Configuration	UPOS	Δ
FST only (no disambig.)	38.96	—
+ Fallback + coverage	∼70	+31
+ AUX/participle map	∼80	+10
+ Viterbi bigram	∼81.20	+1.2
+ Priors + CG-lite	83.92	+2.72
+ Suffix tagger	94.66	+10.74

Table 6: Ablation study (% UPOS, Δ in pp). Values prefixed with ∼ are approximate, measured during incremental development before the final pipeline was frozen. The suffix tagger is a separate statistical component beyond the comonadic pipeline.

5.4 Ablation and Error Analysis

Table 6 shows each component's contribution. The CG-lite rules target top confusion pairs (NOUN/VERB, ADJ/NOUN, ADV/NOUN, PRON/NOUN), each providing deterministic, explainable disambiguation. Structural fixes (+31 pp coverage, +10 pp mapping) contribute more than corpus bigrams (+1.2 pp).

An optional suffix tagger (logistic regression, 157K features, 5 MB) adds 10.74 pp by resolving ambiguities requiring distributional evidence—primarily NOUN/VERB pairs (*tuuli* 'wind' vs. 'blew'), contextual ADJ/NOUN distinctions, and verbal forms following negation verbs where CG rules lack sufficient context. It operates *after* the comonadic pipeline using CG-filtered candidates; it is a separate statistical module. Detailed CG examples are in Appendix B.3.

6 Discussion

The Cofree comonad could provide incremental re-analysis for interactive spell-checking, and the comonad/monad adjunction $W \dashv M$ suggests a connection between comonadic (context-consuming) morphophonology and monadic (effect-producing) generation. We conjecture that every ISL-k function over strings can be expressed as a coKleisli arrow with window size at most $k{+}1$, establishing a formal bridge between subregular phonology (Chandlee and Heinz, 2018) and comonadic composition. The intuition is direct: an ISL-k function's output at each position depends on at most k contiguous input symbols; a coKleisli arrow over a Zipper of the same string has access to the full context but need only examine $k{-}1$ neighbors, matching the ISL sliding-window semantics; `extend` then lifts this local computation to all positions simultaneously. A formal proof requires careful treatment of boundary conditions (string-initial and string-final positions where the Zipper context is truncated) and is our primary direction for future work. We further conjecture that the framework extends to other agglutinative languages: Turkish 4-way vowel harmony, Hungarian ternary harmony, and Estonian consonant gradation all exhibit local context-dependent alternation expressible as coKleisli arrows. The framework should also extend to other length-preserving phenomena: Korean consonant assimilation, Japanese rendaku, North Sámi consonant gradation. Templatic morphology (Semitic root-and-pattern: Arabic K-T-B yields *kataba*, *kitaab*, *maktab*) and reduplication (Indonesian *orang-orang*, Tagalog *bi-bili*) require segment insertion or copying and are not directly captured by the present framework; we develop an extension using an InsertionMap monoid in follow-up work. Verification requires language-specific implementation, which we leave to future work.

7 Related Work

Finite-state morphology. Koskenniemi (1983) established the finite-state paradigm, formalized by Kaplan and Kay (1994) and systematized by Beesley and Karttunen (2003); Hulden (2009) and HFST (Lindén et al., 2013) provide efficient open-source FST tooling. Our contribution is not a faster runtime but a *composition algebra* that keeps rules individually addressable after composition—something FST intersection does not preserve. Our framework operates in the SPE tradition (Chomsky and Halle, 1968), orthogonal to OT's constraint ranking (Prince and Smolensky, 2004).

Functional Morphology. Forsberg and Ranta (2004) treat morphological paradigms as Haskell functions, achieving paradigm compression through functional abstraction. Our framework provides a specific structural guarantee that general functional composition lacks: `extend` uniformly lifts a local rule to all positions with automatic context threading. Forsberg and Ranta (2004) focus on paradigm *generation*; our framework addresses *application* of context-dependent rules, though the MorphGenerator demonstrates that generation also fits naturally.

Neural morphological transducers. Our work targets a different design point—interpretable, offline morphological processing—and the distinct

error profiles of neural vs. rule-based systems suggest hybrid potential (Pirinen, 2019). Recent neural transducers achieve strong results across diverse typologies (see Baxi and Bhatt, 2024, for a survey); SIGMORPHON shared tasks (Kodner et al., 2022; Goldman et al., 2023; Batsuren et al., 2022) continue to drive progress. Combining comonadic rules with neural disambiguation is a natural direction.

Subregular hierarchy. Finnish consonant gradation is input strictly local (ISL); vowel harmony exhibits TSL-2 locality (Heinz, 2018; Chandlee, 2014). Our gradation arrows examine only the immediate left neighbor (ISL); the harmony arrow scans through transparent vowels (matching TSL-2 locality on the vowel tier). The subregular classification could inform arrow design for other languages.

Category theory in NLP. de Felice (2022) applies monoidal categories to compositional semantics; our contribution is orthogonal: *comonadic* structure for context-dependent local transformation.

8 Conclusion

We have shown that the Writer comonad (DeletionSet $\times$ Zipper) restores strict coKleisli compositionality for length-changing morphophonological rules, resolving algebraically a problem that FST cascades address only through multiplicative state explosion—to our knowledge, the first application of the comonad/cellular-automaton correspondence (Capobianco and Uustalu, 2010) to natural language morphophonology.

Several directions for future work emerge: extending the framework to other agglutinative languages (§6); enlarging the coKleisli arrow vocabulary to cover epenthesis and metathesis for a more complete algebraic classification; combining comonadic rules with neural disambiguation to exploit complementary error profiles (Pirinen, 2019); and using the Cofree comonad for incremental re-analysis in interactive spell-checking.

Code and data. The MCE implementation (Rust, Apache-2.0) and evaluation scripts will be released at `https://github.com/yongsk0066/mce` upon publication, with a version tag corresponding to the results reported here. UD Finnish-TDT is distributed under CC-BY-SA 4.0. The Voikko morphological dictionary (voikko-fi) is distributed under GPL-2.0-or-later.

Limitations

The Zipper$\langle char \rangle$ does not encode morpheme boundaries; a Zipper$\langle Morpheme \rangle$ would address this but requires a prior segmentation step. Consonant deletion requires the Writer comonad (§3.6), verified by 44 tests. The 24 CG rule types do not cover the full CG specification's BARRIER conditions and MAP/SUBSTITUTE operations (Karlsson, 1990; Bick, 2000); extending the rule vocabulary is straightforward within the coKleisli framework. The rule-only 83.92% is 12.99 pp below neural TNPP (2.25 pp with the suffix tagger); the gap reflects deep syntactic disambiguation—not a limitation of the comonadic framework itself.

References

Khuyagbaatar Batsuren, Gábor Bella, Aryaman Arora, Viktor Martinovic, Kyle Gorman, Zdeněk Žabokrtský, Amarsanaa Ganbold, Šárka Dohnalová, Magda Ševčíková, Kateřina Pelegrinová, Fausto Giunchiglia, Ryan Cotterell, and Ekaterina Vylomova. 2022. The SIGMORPHON 2022 shared task on morpheme segmentation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 103–116.

Jatayu Baxi and Brijesh Bhatt. 2024. Recent advancements in computational morphology: A comprehensive survey. *arXiv preprint arXiv:2406.05424*.

Kenneth R. Beesley and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI Publications, Stanford, CA.

Eckhard Bick. 2000. *The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Aarhus University Press.

Silvio Capobianco and Tarmo Uustalu. 2010. A categorical outlook on cellular automata. In *Journées Automates Cellulaires (JAC 2010)*, volume 13 of *TUCS Lecture Notes*, pages 88–99. Turku Centre for Computer Science.

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.

Noam Chomsky and Morris Halle. 1968. *The Sound Pattern of English*. Harper & Row, New York.

Giovanni de Felice. 2022. *Categorical Tools for Natural Language Processing*. Ph.D. thesis, University of Oxford.

Markus Forsberg and Aarne Ranta. 2004. Functional morphology. In *Proceedings of the Ninth ACM SIGPLAN International Conference on Functional Programming (ICFP 2004)*, pages 213–223, Snowbird, Utah. ACM.

Omer Goldman, Khuyagbaatar Batsuren, Salam Khalifa, Aryaman Arora, Garrett Nicolai, Reut Tsarfaty, and Ekaterina Vylomova. 2023. SIGMORPHON–UniMorph 2023 shared task 0: Typologically diverse morphological inflection. In *Proceedings of the 20th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 117–125.

Auli Hakulinen, Maria Vilkuna, Riitta Korhonen, Vesa Koivisto, Tarja Riitta Heinonen, and Irja Alho. 2004. *Iso suomen kielioppi*. Suomalaisen Kirjallisuuden Seura, Helsinki. Online edition (2008): http://scripta.kotus.fi/visk. URN:ISBN:978-952-5446-35-7.

Katri Haverinen, Jenna Nyblom, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Anna Missilä, Stina Ojala, Tapio Salakoski, and Filip Ginter. 2014. Building the essential resources for Finnish: The Turku Dependency Treebank. *Language Resources and Evaluation*, 48(3):493–531.

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.

Gérard Huet. 1997. The zipper. *Journal of Functional Programming*, 7(5):549–554.

Mans Hulden. 2009. Foma: A finite-state compiler and library. In *Proceedings of the 12th Conference of the European Chapter of the ACL: Demonstrations (EACL 2009)*, pages 29–32.

Jenna Kanerva, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski. 2018. Turku Neural Parser Pipeline: An end-to-end system for the CoNLL 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 133–142, Brussels, Belgium. Association for Computational Linguistics.

Ronald M. Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

Fred Karlsson. 1990. Constraint grammar as a framework for parsing running text. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING 1990)*.

Jordan Kodner, Salam Khalifa, Khuyagbaatar Batsuren, Hossep Dolatian, Ryan Cotterell, Faruk Akkus, Antonios Anastasopoulos, Taras Andrushko, Aryaman Arora, Nona Atanalov, Gábor Bella, Elena Budianskaya, Yustinus Ghanggo Ate, Omer Goldman, David Guriel, Simon Guriel, Silvia Guriel-Agiashvili, Witold Kieras, Andrew Krizhanovsky, and 11 others. 2022. SIGMORPHON–UniMorph 2022 shared task 0: Generalization and typologically diverse morphological inflection. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 176–203.

Kimmo Koskenniemi. 1983. *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production*. Ph.D. thesis, University of Helsinki. Publication No. 11, Department of General Linguistics.

Krister Lindén, Erik Axelson, Senka Drobac, Sam Hardwick, Juha Kuokkala, Jyrki Niemi, Tommi A. Pirinen, and Miikka Silfverberg. 2013. HFST—a system for creating NLP tools. In *Systems and Frameworks for Computational Morphology*, volume 380 of *Communications in Computer and Information Science*, pages 53–71. Springer.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A light-weight Transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations (EACL 2021)*, pages 80–90. Association for Computational Linguistics.

Dominic Orchard and Alan Mycroft. 2012. A notation for comonads. In *Implementation and Application of Functional Languages (IFL 2012)*, volume 8241 of *LNCS*, pages 1–17. Springer.

Tommi A. Pirinen. 2015. Omorfi—free and open source morphological lexical database for Finnish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 313–315. Linköping University Electronic Press.

Tommi A. Pirinen. 2019. Neural and rule-based Finnish NLP models—expectations, experiments and experiences. In *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages*, pages 104–114, Tartu, Estonia. Association for Computational Linguistics.

Harri Pitkänen. 2006. Hunspell-fi in Kesäkoodi 2006: Final report. Technical report, Finnish Centre for Open Source Solutions (COSS). Available at https://www.puimula.org/htp/archive/kesakoodi2006-report.pdf.

Alan Prince and Paul Smolensky. 2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell, Malden, MA.

Tarmo Uustalu and Varmo Vene. 2005. The essence of dataflow programming. In *Programming Languages and Systems: Third Asian Symposium (APLAS 2005)*, volume 3780 of *LNCS*, pages 2–18. Springer.

A Implementation Details

A.1 Rust and the Absence of Higher-Kinded Types

The formalization in Section 3 is presented in terms of Haskell-style type classes, where a generic `Comonad` class abstracts over any functor W. Rust lacks higher-kinded types (HKTs), so a direct translation of `class Comonad w` is not possible. We therefore adopt a pragmatic approach: the comonad operations are implemented directly on the concrete `Zipper<T>` type, without a generic `Comonad` trait.

```
pub struct Zipper<T> {
    left:  Vec<T>,  // reversed
    focus: T,
    right: Vec<T>,
}

impl<T: Clone> Zipper<T> {
    pub fn extract(&self) -> &T {
        &self.focus
    }

    pub fn extend<U, F>(&self, f: F)
        -> Zipper<U>
    where F: Fn(&Zipper<T>) -> U {
        let focus = f(self);
        let left  = Lefts::new(self)
            .map(|z| f(&z)).collect();
        let right = Rights::new(self)
            .map(|z| f(&z)).collect();
        Zipper { left, focus, right }
    }
}
```

The `Lefts` and `Rights` are internal iterators that produce successive left-shifted and right-shifted zippers by calling `move_left()` and `move_right()` respectively. Each shift clones the internal vectors, resulting in $O(n)$ work per shift and $O(n^2)$ total allocation for a word of length n. For Finnish words (typically 5–15 characters), this cost is negligible; our benchmarks show that a 15-word sentence processes in approximately 0.21 ms. A `SliceZipper` optimization that holds a reference to the underlying slice and moves only a position index would reduce intermediate allocation to $O(n)$.

This approach sacrifices generality—it is not possible to write code that is polymorphic over arbitrary comonads—but gains clarity and avoids the complex trait-bound machinery that Rust's Generic Associated Types (GATs) would require for a full `Comonad` trait. Since our framework requires only a single comonad instance (`Zipper`), this trade-off is appropriate.

A.2 CoKleisli Arrows in Rust

Each morphophonological rule is a plain Rust function with the signature `fn(&Zipper<char>) -> char`. For consonant gradation, the grade parameter is supplied via partial application using a closure:

```
pub fn apply_gradation(
    z: &Zipper<char>, grade: Grade
) -> char {
    let focus = *z.extract();
    let left  = z.peek_left(1).copied();
    let right = z.peek_right(1).copied();
    match find_pattern_at_pos1(
        left, focus, right, grade
    ) {
        Some(pat) => /* target char */,
        None      => focus,
    }
}
// Usage with extend:
let weak = zipper.extend(
    |z| apply_gradation(z, Grade::Weak)
);
```

The CG-lite rules use a trait-based dispatch instead:

```
pub trait CgRule {
    fn apply(&self,
        z: &Zipper<ReadingSet>
    ) -> ReadingSet;
}
```

Each concrete rule type (24 types including `RemoveIfPreceded`, `SelectIfFollowed`, `RemoveByClass`, `SelectByBaseform`, and others) implements this trait. The `apply` method has the coKleisli arrow signature `&Zipper<ReadingSet>` $\rightarrow$ `ReadingSet`, and can be passed directly to extend:

```
let result = zipper.extend(
    |z| rule.apply(z)
);
```

A.3 The Gradation Pattern Table

The 11 consonant gradation patterns are encoded in a static table of `GradationPattern` structures, each specifying a two-character window for both the strong and weak grades:

```
const PATTERNS: &[GradationPattern] = &[
  // Geminates (highest priority)
  GP { s: ['p','p'], w: ['p','\0'] },
  GP { s: ['t','t'], w: ['t','\0'] },
  GP { s: ['k','k'], w: ['k','\0'] },
  // Clusters
  GP { s: ['m','p'], w: ['m','m'] },
  GP { s: ['n','t'], w: ['n','n'] },
  // ... (6 more patterns)
  // Single consonants (lowest priority)
  GP { s: ['\0','p'], w: ['\0','v'] },
  GP { s: ['\0','t'], w: ['\0','d'] },
  GP { s: ['\0','k'], w: ['\0','\0'] },
];
```

The ordering within the table encodes pattern priority: the first matching pattern wins. The `'\0'` character serves as an implementation artifact in the internal representation: at position 0 of single-consonant patterns, it indicates "preceded by any vowel," distinguishing these patterns from cluster patterns that require a specific left neighbor; as

an output character, it signals deletion. When the pattern table produces a `'\0'` output, the Writer comonad arrow (§3.6) intercepts it and records the position in a $DeletionSet$ instead of emitting the null character, restoring strict coKleisli compositionality. The `'\0'` never appears in final output—it exists solely as an internal sentinel consumed by the Writer comonad's materialize step.

A.4 Bidirectional Support

All 11 patterns support both weakening (strong → weak) and strengthening (weak → strong). The find_pattern_at_pos1 function selects which side of the pattern to match against based on the $Grade$ parameter: for weakening, it matches the strong side and produces the weak side; for strengthening, vice versa.

A.5 Test Methodology

The implementation is verified through 317 comonad-related tests organized in six files:

Component	File	#	Verification
Zipper	`zipper`	27	Laws, navigation
Morpho	`finnish`	89	11 patterns, roundtrip
Writer	`writer`	44	Monoid/comonad laws, pipeline equiv.
CG-lite	`cg`	138	24 rule types, safety
Proptest	`proptest`	12	Property-based law verification
Bench	`bench`	7	Perf. regression

Table 7: Test suite organization. All 317 tests pass on CI.

Comonad law tests. The left identity law (extend extract = id) is tested for zippers at multiple focus positions. The right identity law (extract ∘ extend f = f) is tested at every position in a 4-element zipper, confirming that the `Lefts`/`Rights` iterators correctly enumerate all positions. The Writer comonad (`WriterZipper`) is additionally verified for all three comonad laws with monoid accumulation, monoid laws for DeletionSet (identity, associativity, idempotence), and equivalence between the Writer pipeline and the sentinel-based pipeline on all 11 gradation patterns.

A.6 L3′ Proof for the Writer Comonad

We must show $\mathsf{extend}_W\ g \circ \mathsf{extend}_W\ f = \mathsf{extend}_W\ (g \circ \mathsf{extend}_W\ f)$. Expanding the left side, the first extend_W applies f at every position, accumulating a deletion set $D_f = \bigcup_i \pi_1(f(w, z_i))$; the second applies g to the result, accumulating $D_g = \bigcup_j \pi_1(g(w \cup D_f, z'_j))$. The total accumulated set is $(w \cup D_f) \cup D_g$. On the right side, extend_W applies the composed arrow $g \circ \mathsf{extend}_W\ f$ at each position, feeding f's output (including its accumulated deletions) directly into g, yielding $w \cup (D_f \cup D_g)$. Because $(DS, \cup, \emptyset)$ is an associative monoid, $(w \cup D_f) \cup D_g = w \cup (D_f \cup D_g)$, so the deletion components coincide. The character components coincide by L3′ of the underlying Zipper comonad.

Roundtrip tests. For non-destructive gradation patterns (clusters and single consonants), the roundtrip property $\mathsf{strong}(\mathsf{weak}(word)) = word$ is verified. For example, weakening *kampa* to *kamma* and then strengthening back produces *kampa*. The geminate patterns do not satisfy roundtrip because weakening *kaappi* to *kaapi* (via $DeletionSet$ materialization) loses the information that the original contained a geminate; strengthening *kaapi* cannot distinguish between a weakened geminate and a lexical singleton. This asymmetry is a fundamental property of the deletion operation, not a limitation of the comonadic framework.

B Extended Evaluation Details

B.1 Vowel Harmony Verification

Vowel harmony resolution is tested for back-vowel stems (*talo* + *-ssA* → *talossa*), front-vowel stems (*pöydä* + *-ssA* → *pöydässä*), and stems with only neutral vowels (*tie* + *-ssA* → *tiessä*, defaulting to front). The detect_harmony function correctly scans leftward through neutral vowels to find the nearest non-neutral vowel. All three archiphonemes (A, O, U) are tested in both back and front contexts. The harmony function is idempotent: applying it to already-resolved text produces no change.

B.2 Pipeline Integration Examples

The morphophonological pipeline gradation ⪢ harmony ⪢ possessive is tested on words requiring multiple simultaneous transformations:

- *rantAssA* (weak) → *rannassa* (gradation: nt→nn, harmony: A→a)
- *pukussA* (weak) → *puussa* (gradation: k deleted, harmony: A→a)

- *kampAstAVn* (weak) → *kammastaan* (all three rules active)
- *kenkässtAVn* (weak, front) → *kengästään* (gradation: nk→ng, harmony: A→ä, possessive: V→ä)

B.3 CG-lite Disambiguation Examples

The CG-lite module is evaluated on both constructed and corpus-derived Finnish examples. In the rules below, Finnish CG tag names are used: *lukusana* = numeral, *nimisana* = noun, *teonsana* = verb, *laatusana* = adjective, *seikkasana* = adverb.

"kuusi koiraa" ('six dogs'). The word *kuusi* has readings {numeral/kuusi, noun/kuusi}. The rule `SELECT lukusana IF (+1 nimisana)` correctly selects the numeral reading when followed by the noun *koiraa*.

"kuusi kasvaa" ('spruce grows'). With the rule `SELECT nimisana IF (+1 teonsana)`, the noun reading of *kuusi* is correctly selected when followed by the verb *kasvaa*.

"ei voi" ('cannot'). The word *voi* is ambiguous between 'butter' (noun) and 'can' (verb). The rule `SELECT teonsana IF (-1 BASEFORM=ei)` correctly selects the verb reading when preceded by the negation verb *ei*.

3-rule cascade. A sentence position with 4-way ambiguity {noun, verb, adjective, adverb} is reduced to a single reading through three sequential extend passes: (1) `REMOVE laatusana IF (NOT -1 lukusana)`, (2) `REMOVE seikkasana IF (-1 nimisana)`, (3) `SELECT teonsana IF (+1 teonsana)`. This demonstrates the progressive disambiguation enabled by coKleisli composition, where each rule's pruning enables subsequent rules to make sharper decisions.

B.4 Qualitative Coverage Details

The non-roundtrip cases (geminate weakening and *k* deletion) involve information loss: the deletion of a character removes the evidence needed for reverse reconstruction. Of the 11 patterns, 7 exhibit full bidirectional roundtrip correctness; the 4 non-roundtrip cases all involve deletion.

Pattern	In	Out	RT
pp → p	kaappi	kaapi	–
tt → t	matto	mato	–
kk → k	kukka	kuka	–
p → v	tupa	tuva	✓
t → d	katu	kadu	✓
k → ∅	puku	puu	–
mp → mm	kampa	kamma	✓
lt → ll	kulta	kulla	✓
nt → nn	ranta	ranna	✓
rt → rr	parta	parra	✓
nk → ng	kenkä	kengä	✓

Table 8: Gradation patterns as coKleisli arrows (weakening direction). RT = roundtrip $\mathsf{strong}(\mathsf{weak}(w)) = w$.

The Development of Spectral and Temporal Encodings in Speech Sounds

Frank Lihui Tan and **Youngah Do**
The University of Hong Kong
frank.lhtan@connect.hku.hk, youngah@hku.hk

Abstract

This study uses a modeling approach to explore the development of spectral and positional encodings in speech sounds. Humans rely on their auditory system to differentiate between individual sounds in words by analyzing both spectral properties of phonemes and their relative positions. Previous neuroscientific research has identified specific neural populations in the auditory cortex that respond to spectral processing, while behavioral studies have confirmed humans' ability to perceive the relative positions of phonemes in speech sequences. To investigate these encodings, a Long Short-Term Memory (LSTM) autoencoder with a cross-attention mechanism trained on Mel-spectrogram transformed from raw speech data is employed in this research. By conducting ABX tests on the model's representations at various learning stages, we observe the emergence of spectral and positional encodings. The results show that the model excels in distinguishing spectral features similar to neuroscientific findings, and also reveals independent positional encoding through accurate temporal distinctions. Furthermore, we illustrate the developmental trajectory of spectral and positional encodings during the learning process, proposing the need for further investigating their neural correlates.

1 Introduction

When humans process words like "task" or "cast," the auditory system is responsible for identifying individual sounds, e.g., [t], [æ], [s], and [k], as part of the word comprehension. To do so, the auditory system needs to analyze spectral properties of sounds, including their acoustic features (e.g., formants) as well as acoustic-invariant phonemic representations (e.g., [k] in initial and final positions). Previous neuroscientific studies have shown that specific neural populations are selectively responsive to the spectral processing of sounds in auditory cortex, especially the superior temporal gyrus (Chan et al., 2014; Chang et al., 2010; Gwilliams et al., 2022; Keshishian et al., 2023; Mesgarani et al., 2008, 2014; Shestakova et al., 2004). For example, Mesgarani et al. (2014) demonstrated that electrodes placed at different sites within the superior temporal gyrus, each corresponding to distinct neural populations, show selective responses to various phoneme categories, such as plosives, sibilants, vowels, and nasals. Chang et al. (2010) found that when listening to speech stimuli along a continuum, speech representations are categorical and spatially organized in posterior superior temporal gyrus. This organization involves distinct regions which correspond to specific phonetic categories, regardless of their acoustic variations. Not only do populations of neurons exhibit selective responses, but individual neurons have also been found to respond to particular phonemes or phonemic categories, such as vowels versus consonants, and even to their phonological patterns or statistical regularities (Lakretz et al., 2021; Leonard et al., 2024). The development of such spectral encodings of sounds begins early in human development, within the first year of life (Di Liberto et al., 2023). In parallel, computational modeling studies have demonstrated that neural networks trained on raw speech can spontaneously learn phoneme-like representations in their latent spaces (e.g. Baevski et al., 2020; Beguš, 2020).

In addition to understanding spectral characteristics of sounds, it is also crucial to recognize the relative positions of sounds, such as [t-æ-s-k] or [k-æ-s-t], in order to successfully comprehend words. Humans' auditory system is capable of positional encoding of sounds, allowing the detection of sound sequences with intervals as short as 15–20 ms (Hirsh, 1959; Steinschneider, 2005). Behavioral studies showed that humans can detect the relative positions of phonemes in speech streams (Benavides-Varela and Mehler, 2015; Endress and Mehler, 2010; Fló, 2021; Zhang et al.,

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 113–126
July 3-4, 2026 ©2026 Association for Computational Linguistics

2003). For example, Fló (2021) conducted a sequence perception study in which listeners were trained on novel words. Although all syllables in the test words were familiar, listeners preferred words that maintained some syllables in their familiarized serial positions over words where all syllables had changed position, indicating that humans can track phonemes' relative positions. Similarly,
to develop emergent positional awareness, enabling them to perform counting-like and order-sensitive tasks without explicit positional encoding (Gers and Schmidhuber, 2001; Weiss et al., 2018).

While there is a range of evidence supporting the positional encoding of phonemes in humans, it is still unclear whether distinct neural encodings are specifically dedicated to marking the timestamps of phoneme sequences. This idea is akin to the spectral encoding found in Gwilliams et al. (2022), where specific neural encoding of spectral information of phonemes was identified. Neuroscientific studies have shown that neural populations respond differently to the same sound depending on its relative position within a sequence Leonard et al. (2015): for example, an electrode with a strong preference for the /n/ sound at the syllable-initial position showed no specific preference when /n/ appeared at the syllable-final position, suggesting that temporal context influences neural responses to speech sounds. However, there is currently no conclusive evidence for the existence of neural mechanisms that explicitly encode the positional information of phonemes independently of their spectral features. In the current study, the first goal is to investigate whether a positional encoding of phonemes, similar to spectral encoding, can be observed during speech perception. The second goal is to trace the learning trajectories by which learners acquire spectral and positional encodings of phonemes. As shown, previous research has provided diverse evidence supporting the existence of spectral and positional encodings of phonemes in human cognition, but the process by which such encodings are established in learners remains unclear, including whether similar patterns are observed in the development of both spectral and positional encodings. To our knowledge, no longitudinal data exist which directly compare the emergence and evolution of spectral and positional encodings. In this context, neural networks offer a promising framework for investigating the developmental trajectory of spectral and positional encoding in the acquisition of speech sounds and their results may help formulate hypotheses about analogous neural processes in human perceptual development.

The current study uses a modeling approach to replicate spectral and positional encodings observed in humans and investigate how these two different types of encodings evolve and interact over time. To do so, we simulate their learning process using a Long Short-Term Memory (LSTM) autoencoder equipped with cross-attention mechanism. The autoencoder is trained on Mel-spectrogram transformed from raw speech data, in order to examine the emergence and evolution of these encodings during language acquisition. We conduct ABX tests on the model's representations from various layers to assess the information captured in different stages of learning. Through this experiment, we expect to observe spectral encodings, consistent with previous neuroscientific findings. If the model successfully replicates humans' positional encoding abilities of phonemes, we also anticipate the emergence of independent positional encoding, which should be observed from the ABX distinction performance on positional contrasts, e.g., the same phoneme in different positions. In terms of developmental progression, we expect the model's ability to distinguish spectral and positional features to improve steadily across training epochs.

2 Methods

The model was trained to understand and reconstruct sound sequences accurately. Its ability to reconstruct the input was evaluated through a reconstruction task, measuring how well it replicated the original data. The model was expected to capture the sequential patterns effectively and accurately reproduce the input during the reconstruction process.

2.1 Model Architecture

An LSTM-based deep autoencoder model (Saini and Singh, 2024) was employed to learn auditory input by processing it sequentially without explicit guidance. The overall architecture follows the same design as that adopted in Tan and Do (2025). As there is no explicit feedback or learning guidance, the learning process is similar to how infants learn through passive listening. As in Figure 1, the model consists of an encoder, decoder, and cross-attention mechanism, all of which were designed to capture

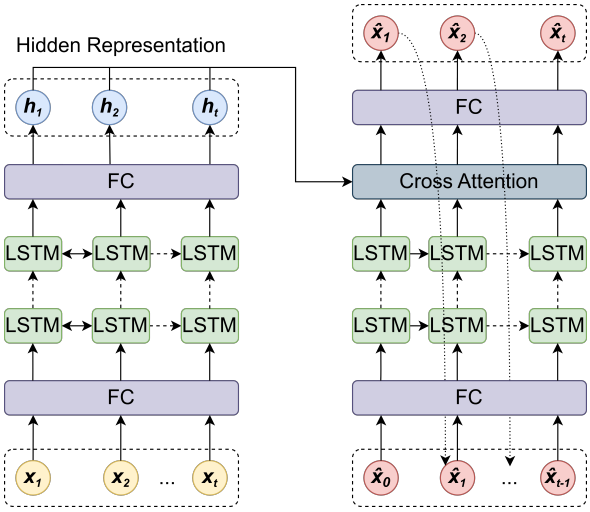


Figure 1: Illustration of model architecture.

spectral information as well as the temporal dependencies of the auditory sequences. The encoder and decoder were built with fully connected layers (FC) and LSTM layers. FC layers connect every neuron in one layer to every neuron in the next layer, enabling the learning of relationships between features in the data. LSTM layers capture sequential dependencies of the data. Cross-attention, implemented as scaled dot-product attention (Vaswani et al., 2017), was included to ensure effective information flow between the encoder and decoder. See Appendix A for details.

2.2 Dataset Construction and Preprocessing

The "train-clean-100" subset from the LibriSpeech corpus (Panayotov et al., 2015) was used, containing 100 hours of 16kHz English speech recorded by 251 speakers (125 female and 126 male). We relied on existing transcriptions (Lugosch et al., 2019) generated by the Montreal Forced Aligner (McAuliffe et al., 2017) to produce sub-word clusters from continuous recordings of sentences. We adopted a segment-based concatenation to construct the stimuli rather than extracting naturally occurring VCV sequences from continuous speech. Segments were extracted as individual phoneme realizations from the corpus and then artificially concatenated to construct VCV tokens following the templates VSV and VSHV, rather than being taken from naturally occurring continuous VCV sequences. The concatenation process ensured within-token speaker consistency such that each concatenated VCV token was constructed exclusively from segments produced by a single speaker. We selected from the English phoneme inventory six vowels (four monophthongs: AA, AE, IY, UW; and two diphthongs: AY, EY) and two consonants (S and SH). In total, this process yielded 66,792 V-S-V and around 25,238 V-SH-V sequences. We randomly selected a subset of 25,238 V-S-V sequences to match the size of the V-SH-V dataset. Mel-spectrograms were then extracted from the audio recordings and were standardized with mean and variance normalization.

2.3 Training

The model was trained using the Adam optimizer with a learning rate of 0.0005 (Kingma and Ba, 2014). To reduce the impact of random variation, each experiment was repeated five times independently, with the model trained for 100 epochs per run to ensure convergence. Performance was recorded at every epoch, including epoch 0, which represents the model's untrained state before any learning has taken place.

2.4 Layers and Evaluation Metrics

To evaluate the encoding capabilities of the model, we conducted ABX tests on outputs from all layers (Millet and Dunbar, 2020; Nguyen et al., 2020; Schatz et al., 2013, 2018). All representations for the tests were normalized using z-score normalization prior to distance computation. To reduce random variation, each ABX test was conducted six times, with 20 samples drawn from each phoneme class for every run.

All intermediate representations of the model were tested with the ABX test. The 96-dimensional Mel-spectrogram of the input signal was included as a baseline and labelled as *input-mel-spectrogram*. The final-layer outputs from the bidirectional encoder LSTM were sepatated into forward and backward representations and labelled as *encoder-LSTM-last-forward/backward-output*, respectively. The output of the final FC layer of the encoder was labelled as *encoder-output*. In the decoder, the representation produced after cross-attention was labelled as *cross-attention-output*. This layer-by-layer evaluation allows us to trace how phonological information is processed, refined, and abstracted as it propagates through the model during learning.

For each layer, we conducted two types of ABX tests: phoneme contrast test and phoneme position test. The phoneme contrast ABX test assessed the model's ability to distinguish between phoneme categories, e.g., /p/ vs. /k/ or /i/ vs. /u/. Good

performance in the phoneme contrast ABX test indicates that the corresponding layer is sensitive to phonemic or spectral contrasts.

The phoneme position ABX test, on the other hand, evaluated the model's ability to distinguish the same phoneme at different positions within the sequence (e.g., the first versus the third /i/ in the constructed VCV structure). Specifically, ABX triplets were constructed such that A and B corresponded to the same vowel category realized at two different positions in the VCV structure. Good performance in this test suggests that the layer is sensitive to positional information independent of spectral differences.

As a control test, the distinction between the same phoneme at the same position across different tokens was examined. For each training epoch, including epoch 0, we collected the model's outputs on the evaluation dataset and conducted the aforementioned ABX tests.

3 Results

We present our findings from two main aspects. First, we report model's ability to learn phonemic contrast (spectral encoding) and phoneme position contrast (positional encoding) and show how performance differs across layers. Next, we explore how the two encoding capabilities of different layers evolve over the course of training.

3.1 Division of Codes: Final Epoch Distinction Performances

To evaluate whether the model developed distinct spectral and positional encodings, the ABX test results from the final epochs were taken as the model's learning outcomes during the later stages of training. Specifically, ABX error rates were collected from epochs 95 to 100, as these represent the concluding phases of training, offering a reliable reflection of the model's final performance. The inclusion of five consecutive epochs helped reduce the impact of random fluctuations. While all hidden dimensions were assessed and showed similar trends (see OSF), a mid-range value of 16 was chosen for presentation purposes.

The results are depicted in five representations as shown in Figure 2: (a) Input Mel-Spectrogram illustrating the initial state, (b-c) two-directional last encoder outputs prior to forming the encoder output, (d) encoder output, which represents the final knowledge encapsulated by the encoder, and (e) cross-attention output representing a near-final representation in the decoder, which is used for reconstruction.

The input Mel-spectrogram (a) clearly distinguished between phoneme categories, with a low mean ABX error rate of 0.177, but struggled to encode phoneme position information, resulting in high error rates of 0.482. This implies that while the input data exhibited distinct phonemic contrasts (e.g., /i/ vs. /u/), positional distinctions between phonemes were less evident in the input (e.g., initial /i/ vs. final /i/). In the intermediate LSTM layers, the performance trends remained consistent with those of the input Mel-spectrogram, showing better accuracy in distinguishing phoneme category differences than phoneme position differences (see OSF for the details). Upon further data processing, the last LSTM layers (b-c) revealed a significant difference in how they each encoded spectral and positional elements within the forward and backward representations. Specifically, the forward output (b) excelled in capturing phoneme position details with a low error rate of 0.098 but struggled to differentiate phoneme categories, resulting in a high error rate of 0.476. Conversely, the backward output (c) demonstrated strong encoding of phoneme category distinctions with relatively low error rates (0.183) but encountered challenges in assessing phoneme position, showing high error rates (0.476). The two types of outputs, thus, demonstrated complementary distributions in encoding spectral and positional representations.

The encoder output (d), i.e., the learned outcome at the final stage of the encoding process, showed the lowest average error rates, indicating a successful learning outcome at the final stage of the encoder. This layer effectively integrated spectral and positional information, achieving moderate error rates for phoneme category differentiation (0.303) and low error rates for phoneme position distinction (0.105). Nevertheless, when cross-attention was applied to this encoder output in the decoding process, the representation in the decoder reverted to emphasizing spectral encoding, a different encoding type compared to the final encoder stage. The output from the cross-attention layer exhibited the lowest error rates in phoneme category classification among all layers (0.125) but faced challenges in preserving phoneme position information (0.485).

The ABX test results show a combination of spectral and positional encoding. Notably, layers within the model specialize in either spectral or

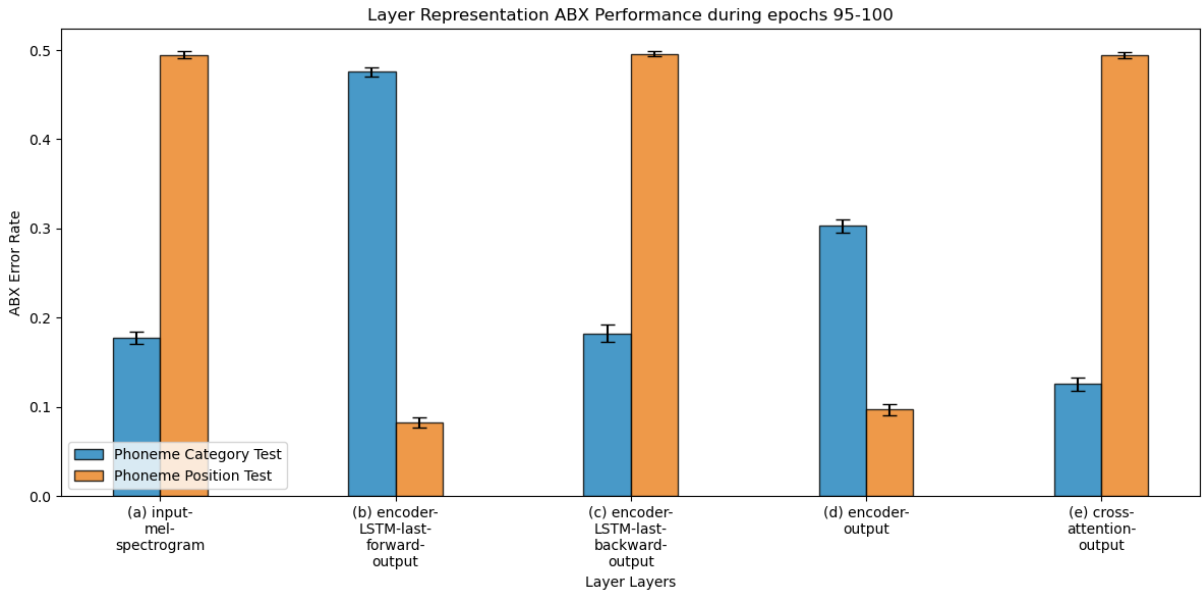


Figure 2: Layer representations' ABX test error rate on phoneme category test and phoneme position test for epochs 95-100.

positional information encoding to a certain extent. This relatively specialized division among layers could allow the model to optimize the reconstruction of input data by assigning specific encoding tasks to different layers. Specifically, successful spectral encoding in the input Mel-spectrogram is expected, as Mel-spectrogram features represent primarily spectral information; consequently, the same phoneme occurring in different positions yields highly similar representations, leading to weak positional distinctions in ABX comparisons. For the encoder output, it can also be inferred from its bridging role between the encoder and decoder that the encoder output encodes both spectral and positional information. The specialization of the cross-attention output in spectral encoding can be understood from its structural role—positioned before the final FC layer, it prioritizes spectral information to optimize reconstruction, mirroring the input. In contrast, the separate encoding of spectral and positional information in the final LSTM outputs, and their relation to forward and backward directions, is not easily explained solely by the model's structure. While the exact reasons for why and how each direction is linked to its specific encoding, i.e., forward output for spectral encoding and backward output for positional encoding, are not entirely predictable, it is evident that there is a clear division of labor based on their respective directions. We will revisit this topic in the Discussion section.

3.2 Model Development: Evaluation along Training Trajectory

The developmental trajectories of ABX error rates provide evidence of how spectral and positional information encoding in layers evolved during training. We examine how different layers progressed from an initial stage to their eventual specialization in encoding spectral or positional information.

As shown in Figure 3, the model initially struggled to distinguish between phoneme categories and phoneme positions, with middle to high ABX error rates across all layers. Following the earlier epochs, the L-shaped learning trajectories were found from some layers, both for phoneme contrast (solid lines) and phoneme position tests (dotted lines), with phoneme position tests showing more rapid learning. They are characterized by initial high ABX error rates, followed by a steady decline (comparison of ABX error rates between epoch 0 and the epoch with the lowest error rate, $p < 0.001$) as the model improves its ability to detect phoneme contrast or position, then stabilizes at a lower error rate with comparatively minor fluctuations. This pattern is primarily observed in layers that strongly encode either phoneme category or phoneme position in the final epochs, as described in the previous section, e.g., the encoder output. This pattern distribution suggests that each layer exhibits L-shaped learning trajectories for the specific type of information they specialize in encoding. For example, in the phoneme position test, the encoder LSTM last

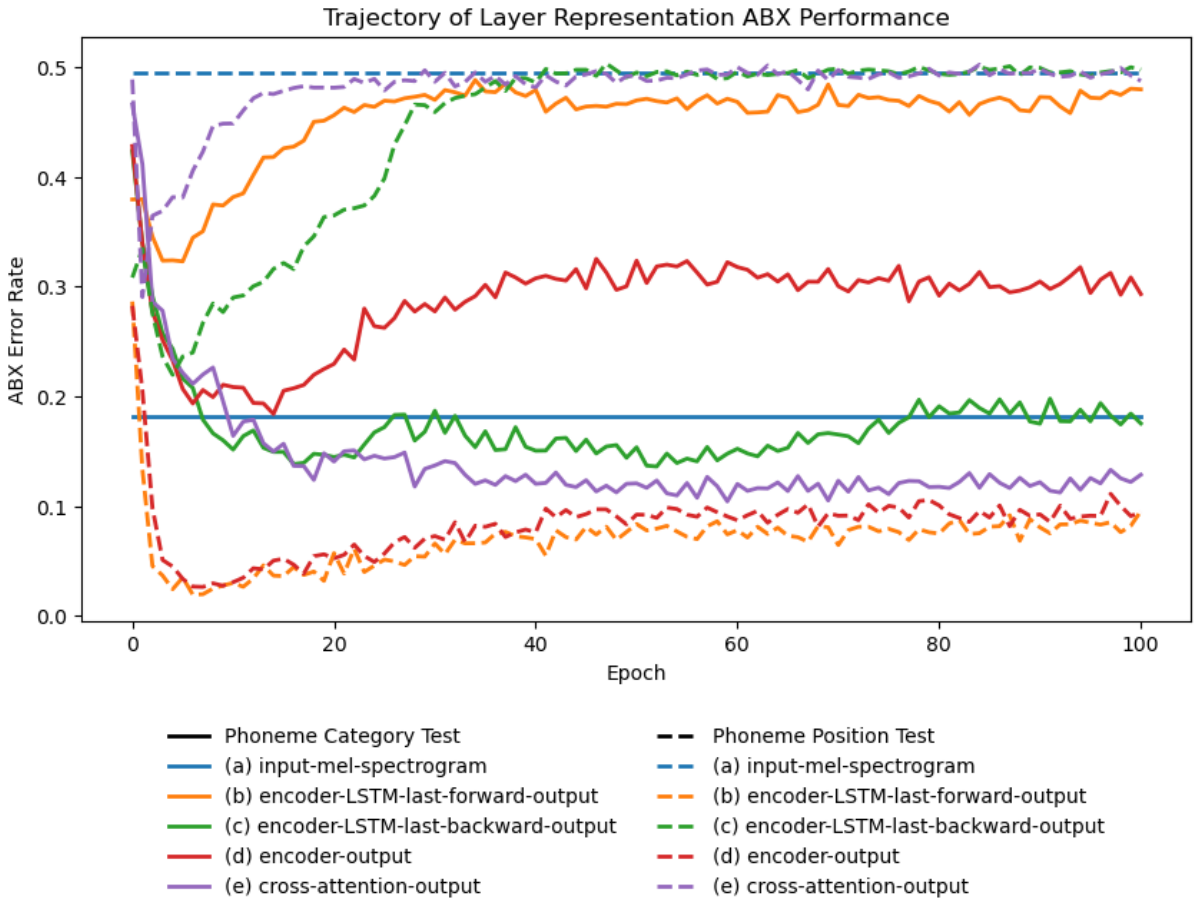


Figure 3: Trajectory of layer representations' ABX test error rate on phoneme category test (solid lines) and phoneme position test (dotted lines).

forward output, and the encoder output followed an L-shaped trajectory, demonstrating steady improvement over training. Similarly, in the phoneme category test, this pattern was observed consistently in the encoder LSTM last backward output and the cross-attention output.

However, not all learning trajectories followed a straightforward progression. As shown in Section 3.1, most layers ultimately specialize in encoding only one type of information, either spectral or positional. Accordingly, each layer typically exhibits only one L-shaped trajectory, corresponding to the information type it retains. The other trajectory does not remain flat at chance level; rather, it often shows an initial rise in performance before regressing. For instance, the encoder LSTM last backward output and the cross-attention output, which specialize in spectral encoding, initially displayed some positional encoding ability. Yet, unlike the trajectories that steadily improved in their specialized information, these non-specialized trajectories showed early progress followed by decline, resulting in the loss of the transiently encoded information. This phenomenon, evidenced by significant differences in ABX error rates between epoch 0, the final epoch, and the epoch with the lowest error rate ($p < 0.001$ for both comparisons), may suggest an "exploratory" learning strategy. Rather than having predefined roles from the outset, each layer underwent an adaptive process, eventually converging on the specific type of information it would specialize in encoding. These findings indicate that the specialization of layers observed in the final epoch did not emerge immediately but rather resulted from a gradual process of trial and refinement. Instead of encoding a single type of information from the start, many layers initially attempted to encode both spectral and positional details before settling into distinct roles.

4 Discussion

4.1 Summary of Main Findings

This study addressed two primary research questions: (1) whether independent positional encod-

ings exist to mark the timestamps of sound sequences separately from spectral representations and (2) how spectral and positional encodings develop over time in neural network learners newly introduced to language. As positional understanding is essential for perceiving phonemes in a sound stream (Chambers et al., 2003; Saffran et al., 1996), and behavioral studies have provided evidence supporting humans' ability for positional encoding of phonemes (Benavides-Varela and Mehler, 2015; Endress and Mehler, 2010; Fló, 2021; Zhang et al., 2003), we hypothesized that an independent positional encoding mechanism would be observed as a parallel to the spectral encoding observed by Gwilliams et al. (2022). We further anticipated a steady improvement in both spectral and positional encodings along with their separation across different layers of the model.

By evaluating the model's different layers on the phoneme contrast test (spectral knowledge) and the phoneme position test (positional knowledge), we found that the model successfully learned distinct encodings of phoneme categories (spectral knowledge) and phoneme positions (positional knowledge), with different layers specializing in each task. Moreover, the trajectory of different layers' outputs showed how such spectral and positional encodings evolved during the early stages of learning. We show that most layers exhibited expected learning trajectories showing a specialized type of information, either spectral or positional, and they steadily improved their specialized knowledge over time. However, as to the opposite information that a certain layer is not successfully encoding (e.g., spectral information for layers specialized in positional encoding and vice versa), instead of consistently displaying low distinction, these layers initially encoded both spectral and positional distinctions before gradually suppressing the encoding of the two types of information as training progressed. We elaborate each point below.

4.2 Phoneme Category and Position Encoding Specialization

The observed specialization of layers aligns with previous neuroscientific research demonstrating the existence of spectral encoding, i.e., position-invariant neural phoneme encoding (Gwilliams et al., 2022), as it supports a designated space for encoding the spectral information of phonemes. For example, the encoder LSTM last forward layer encoded the positional information of phonemes within a sequence while largely disregarding their spectral characteristics. This pattern suggests that positional encoding operates relatively independently from phonemic categorization. Literature on positional encoding suggests that its mechanisms are not restricted to language; Instead, positional encoding has also been identified in studies on non-linguistic sound sequences (Furl et al., 2011; Overath et al., 2007; Skerritt-Davis and Elhilali, 2018). For instance, Furl et al. (2011) observed that neural responses to temporal sequences of pure, non-linguistic tones were stronger when the sequence violated expectations based on prior training. Similarly, Overath et al. (2007) found that pure tone sequences with higher entropy led to increased neural activity in the planum temporale. Similar encoding of temporal patterns and sequence statistics extends beyond humans, as evidenced by animal auditory perception as well (Bouchard and Brainard, 2013; Schnupp et al., 2006), highlighting the broader significance of identifying temporal encoding of sounds. Additionally, behavioral and neural studies on object sequence recall in animals further support the existence of relative position sensitivity beyond human language and auditory perception (Ninokura et al., 2003, 2004; Orlov et al., 2000). Ninokura et al. (2004) identified neural populations in the dorsal lateral prefrontal cortex that selectively responded to relative position of objects, regardless of their physical properties. Such a diverse range of evidence beyond phonemes and human perception suggests that the positional encoding mechanisms identified in the current study have the potential to offer insights into the broader cognitive processes involved in auditory perception and temporal pattern recognition beyond language.

Moreover, Ninokura et al. (2004) discovered neural populations that encode objects' identity as well as relative position, with some neurons specialized in each property individually. This indicates that combined encoding arises through the interaction of separate yet complementary representations. Our findings on the encoder output, which merges spectral and positional information, align with those observed in object perception as in Ninokura et al. (2004). Instead of inherently encoding both spectral and positional properties from the start, the encoder output, representing the learned knowledge of the encoder, construct a joint representation by transforming independently learned spectral and positional encodings. This mechanism may represent a broader principle within the

human auditory system, where distinct neural populations process various aspects of speech before integrating their outputs at higher processing levels. This organization aligns with the structure proposed in the Phonological Loop Model (Burgess and Hitch, 1999), which posits that separate item property and order processing mechanisms contribute to sequence perception and recall.

4.3 Developmental Trajectories of Encoding Specialization

The analysis of the model's learning trajectory revealed patterns that align with previous research on phoneme acquisition. The L-shaped learning trajectory observed in spectral encoding—particularly in layers such as the encoder LSTM last backward layer and the cross-attention layer—indicates a progressive improvement in phonemic encoding during the early stages of training. This trend mirrors findings in human development, where phonemic representations become increasingly detailed and acoustically invariant over time. For instance, Di Liberto et al. (2023) demonstrated that infants in their first year of life gradually refine their phoneme encoding, developing more robust and stable phonemic categories. Similarly, behavioral and neural research supports the notion that infants' ability to distinguish phonemes in their native language steadily improves as they are exposed to linguistic input (Cheour et al., 1998; Kuhl et al., 2006; Rivera-Gaxiola et al., 2005; Sundara et al., 2006), similar to the L-shaped developmental trajectory in the present study. A similar developmental trajectory was observed in layers specializing in positional encoding as well, such as the encoder LSTM last forward layer and the encoder output layer.

Furthermore, our analysis revealed that positional encoding developed more rapidly than spectral encoding, reaching a stable level of proficiency earlier in training. This pattern also aligns with developmental findings in infants, where sensitivity to temporal structure emerges before robust phoneme discrimination. Studies have shown that newborns already exhibit positional awareness (Gervain et al., 2008), and by five months of age, the better learners among infants begin to show sensitivity to relative positions (Fló, 2021). This sensitivity continues to develop, with evidence of positional awareness at seven months (Benavides-Varela and Mehler, 2015) and nine months of age (Gerken, 2006). In contrast, while infants display an early universal sensitivity to phonemes, the ability to reliably distinguish phonemic contrasts in their native language develops more gradually and continues to refine over time (Cheour et al., 1998; Kuhl et al., 2006; Rivera-Gaxiola et al., 2005; Sundara et al., 2006). The developmental trajectory of the current model reflects a similar pattern, in which positional encoding emerges earlier, likely supporting later refinements in phoneme representation.

The learning trajectories of spectral and positional encoding indicate that different layers specialize in encoding distinct types of information, either spectral or positional. Despite the anticipation that layers would not encode information opposite to their specialization, our results unveil that many layers initially encode both types of information. For instance, the encoder LSTM last forward layer initially showed some capacity for encoding phonemic contrast, even though this layer was specialized for positional encoding. Similarly, the encoder LSTM last backward layer exhibited a certain degree of positional encoding at first, despite specializing in spectral encoding. This suggests that layers were not strictly selective for the type of information to encode innately but rather engaged in an exploratory phase where they encoded both spectral and positional information to some extent. However, as learning progressed, rather than continuously improving in encoding both types of information or simply maintaining the initial level of encoding for the non-specialized dimension, the layers actively suppressed the encoding of the non-specialized information. They restructured their representation space and de-learned certain contrasts. This trajectory suggests that specialization was not predefined but emerged dynamically from exploratory learning as the network optimized its representations for the reconstruction task. Instead of having rigidly assigned roles from the beginning, each layer initially explored how to encode both spectral and positional information before gradually refining its specialization.

4.4 Limitations

While our findings align with previous neural and behavioral studies, several limitations should be acknowledged. First, we presented modeling evidence supporting positional encoding and traced the model's developmental trajectories for both spectral and positional encoding. To our knowledge, no empirical neuroscientific evidence currently suggests the presence of distinct neurons or neural groups dedicated to positional encoding in

speech perception, nor are there longitudinal studies comparing the emergence and development of spectral and positional encoding in humans. Yet, the LSTM autoencoder used here is not intended as a mechanistic model of infant auditory cortex, and we do not assume that human learners use the same architecture, objective function, or optimization procedure. Rather, consistent the view that cognitive models are tools for exploring the implications of theoretical ideas under simplified and explicitly specified conditions (McClelland, 2009), the present model serves as a controlled artificial learner. In this sense, the contribution of the model does not depend on the claim that human infants learn in the same implementational manner as the network. Instead, its relevance lies in providing a computational demonstration: it tests whether a particular representational organization can emerge from the information structure of sequential speech under the pressure to preserve that information for reconstruction. The current results therefore show that a division of labor between phoneme-category information and relative-position information is a computationally plausible outcome for an artificial learner exposed to sequential speech input. Given that human learners are also sensitive to statistical regularities in auditory input (Saffran et al., 1996; Saffran and Kirkham, 2018), this modeling result offers a hypothesis for future empirical work: under similar information-structural constraints, human learners may also arrive at separable phoneme-category and relative-position representations through a comparable developmental trajectory. While neurophysiological evidence can inspire modeling studies, we advocate for reciprocal influence, where modeling data could also generate hypotheses and inspire neurophysiological research. We hope that our findings can stimulate neuroscientists and acquisitionists to investigate parallels between spectral encoding and positional encoding.

Second, we did not test alternative model architectures, such as Transformer-based models or convolutional networks. The emergence of positional encoding in our LSTM-based model may be inherently linked to the sequential nature of recurrent networks, which are explicitly designed for processing temporal dependencies. However, we argue that using an LSTM was the appropriate choice in the current study, given that sequential processing is a fundamental and early-acquired cognitive ability in humans. Research has shown that even newborns exhibit increased neural responses to reduplicated sequences (Gervain et al., 2008) and demonstrate sensitivity to ordinal positions by six months of age (Lewkowicz, 2013). Additionally, sequential learning plays a crucial role in linguistic processes, such as phonotactic pattern learning (c.f. Saffran and Kirkham, 2018), which infants begin acquiring as early as seven months (Thiessen and Saffran, 2003). Future work should explore whether similar spectral-positional encoding specializations emerge in non-recurrent architectures to determine if this effect is unique to LSTMs or a more general feature of neural network-based learning.

Finally, we want to note that the observed over-learning and subsequent de-learning trajectory of non-target information lacks empirical support from human neurophysiological studies. Nevertheless, our findings have shown that layers specializing in spectral or positional encoding followed learning trajectories similar to those observed in phoneme acquisition and positional learning research in the field. Therefore, we argue that the observed patterns in the current study warrant further neuroscientific investigation. A deeper understanding of how neural systems transition from broad, exploratory encoding to specialized representations may provide valuable insights into both human cognitive development and artificial learning architectures.

5 Conclusion

In conclusion, this study used an LSTM-based autoencoder to examine how spectral and temporal encodings can emerge and specialize during learning. By applying layer-wise ABX analyses, we showed that different components of the model come to preferentially encode phoneme category or phoneme position information, while higher-level representations integrate both. Yet, these encoding specializations were not present from the start but emerged gradually through learning.

Code Availability

Data, analysis scripts, and all results can be found at the following OSF link: https://osf.io/pja86/?view_only=3d182a56f7bb46e796149af313516bc2.

References

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. *arXiv preprint*. Version Number: 3.

Gašper Beguš. 2020. Generative Adversarial Phonology: Modeling Unsupervised Phonetic and Phonological Learning With Neural Networks. *Frontiers in Artificial Intelligence*, 3:44.

Silvia Benavides-Varela and Jacques Mehler. 2015. Verbal Positional Memory in 7-Month-Olds. *Child Development*, 86(1):209–223.

Kristofer E. Bouchard and Michael S. Brainard. 2013. Neural Encoding and Integration of Learned Probabilistic Sequences in Avian Sensory-Motor Circuitry. *The Journal of Neuroscience*, 33(45):17710–17723.

Neil Burgess and Graham J. Hitch. 1999. Memory for serial order: A network model of the phonological loop and its timing. *Psychological Review*, 106(3):551–581.

Kyle E. Chambers, Kristine H. Onishi, and Cynthia Fisher. 2003. Infants learn phonotactic regularities from brief auditory experience. *Cognition*, 87(2):B69–B77.

Alexander M. Chan, Andrew R. Dykstra, Vinay Jayaram, Matthew K. Leonard, Katherine E. Travis, Brian Gygi, Janet M. Baker, Emad Eskandar, Leigh R. Hochberg, Eric Halgren, and Sydney S. Cash. 2014. Speech-Specific Tuning of Neurons in Human Superior Temporal Gyrus. *Cerebral Cortex*, 24(10):2679–2693.

Edward F. Chang, Jochem W. Rieger, Keith Johnson, Mitchel S. Berger, Nicholas M. Barbaro, and Robert T. Knight. 2010. Categorical speech representation in human superior temporal gyrus. *Nature Neuroscience*, 13(11):1428–1432.

Marie Cheour, Rita Ceponiene, Anne Lehtokoski, Aavo Luuk, Jüri Allik, Kimmo Alho, and Risto Näätänen. 1998. Development of language-specific phoneme representations in the infant brain. *Nature Neuroscience*, 1(5):351–353.

Giovanni M. Di Liberto, Adam Attaheri, Giorgia Cantisani, Richard B. Reilly, Áine Ní Choisdealbha, Sinead Rocha, Perrine Brusini, and Usha Goswami. 2023. Emergence of the cortical encoding of phonetic features in the first year of life. *Nature Communications*, 14(1):7789.

Ansgar D. Endress and Jacques Mehler. 2010. Perceptual constraints in phonotactic learning. *Journal of Experimental Psychology: Human Perception and Performance*, 36(1):235–250.

Ana Fló. 2021. Evidence of ordinal position encoding of sequences extracted from continuous speech. *Cognition*, 213:104646.

Nicholas Furl, Sukhbinder Kumar, Kai Alter, Simon Durrant, John Shawe-Taylor, and Timothy D. Griffiths. 2011. Neural prediction of higher-order auditory sequence statistics. *NeuroImage*, 54(3):2267–2277.

LouAnn Gerken. 2006. Decisions, decisions: infant language learning when multiple generalizations are possible. *Cognition*, 98(3):B67–74.

F.A. Gers and E. Schmidhuber. 2001. LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks*, 12(6):1333–1340.

Judit Gervain, Francesco Macagno, Silvia Cogoi, Marcela Peña, and Jacques Mehler. 2008. The neonate brain detects speech structure. *Proceedings of the National Academy of Sciences of the United States of America*, 105(37):14222–14227.

Laura Gwilliams, Jean-Remi King, Alec Marantz, and David Poeppel. 2022. Neural dynamics of phoneme sequences reveal position-invariant code for content and order. *Nature Communications*, 13(1):6606.

Ira J. Hirsh. 1959. Auditory Perception of Temporal Order. *The Journal of the Acoustical Society of America*, 31(6):759–767.

Jeff Hwang, Moto Hira, Caroline Chen, Xiaohui Zhang, Zhaoheng Ni, Guangzhi Sun, Pingchuan Ma, Ruizhe Huang, Vineel Pratap, Yuekai Zhang, Anurag Kumar, Chin-Yun Yu, Chuang Zhu, Chunxi Liu, Jacob Kahn, Mirco Ravanelli, Peng Sun, Shinji Watanabe, Yangyang Shi, and 5 others. 2023. TorchAudio 2.1: Advancing speech recognition, self-supervised learning, and audio processing components for PyTorch. *arXiv preprint*. ArXiv:2310.17864.

Menoua Keshishian, Serdar Akkol, Jose Herrero, Stephan Bickel, Ashesh D. Mehta, and Nima Mesgarani. 2023. Joint, distributed and hierarchically organized encoding of linguistic features in the human auditory cortex. *Nature Human Behaviour*, 7(5):740–753.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. Version Number: 9.

Patricia K. Kuhl, Erica Stevens, Akiko Hayashi, Toshisada Deguchi, Shigeru Kiritani, and Paul Iverson. 2006. Infants show a facilitation effect for native language phonetic perception between 6 and 12 months. *Developmental Science*, 9(2).

Yair Lakretz, Ori Ossmy, Naama Friedmann, Roy Mukamel, and Itzhak Fried. 2021. Single-cell activity in human STG during perception of phonemes is organized according to manner of articulation. *NeuroImage*, 226:117499.

Matthew K. Leonard, Kristofer E. Bouchard, Claire Tang, and Edward F. Chang. 2015. Dynamic Encoding of Speech Sequence Probability in Human

Temporal Cortex. *The Journal of Neuroscience*, 35(18):7203–7214.

Matthew K. Leonard, Laura Gwilliams, Kristin K. Sellers, Jason E. Chung, Duo Xu, Gavin Mischler, Nima Mesgarani, Marleen Welkenhuysen, Barundeb Dutta, and Edward F. Chang. 2024. Large-scale single-neuron speech sound encoding across the depth of human cortex. *Nature*, 626(7999):593–602.

David J. Lewkowicz. 2013. Development of ordinal sequence perception in infancy. *Developmental Science*, 16(3):352–364.

Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. 2019. Speech Model Pre-training for End-to-End Spoken Language Understanding.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi. In *Proc. Interspeech 2017*, pages 498–502.

James L. McClelland. 2009. The Place of Modeling in Cognitive Science. *Topics in Cognitive Science*, 1(1):11–38.

Nima Mesgarani, Connie Cheung, Keith Johnson, and Edward F. Chang. 2014. Phonetic Feature Encoding in Human Superior Temporal Gyrus. *Science*, 343(6174):1006–1010.

Nima Mesgarani, Stephen V. David, Jonathan B. Fritz, and Shihab A. Shamma. 2008. Phoneme representation and classification in primary auditory cortex. *The Journal of the Acoustical Society of America*, 123(2):899–909.

Juliette Millet and Ewan Dunbar. 2020. The Perceptimatic English Benchmark for Speech Perception Models. *arXiv preprint*.

Tu Anh Nguyen, Maureen de Seyssel, Patricia Rozé, Morgane Rivière, Evgeny Kharitonov, Alexei Baevski, Ewan Dunbar, and Emmanuel Dupoux. 2020. The Zero Resource Speech Benchmark 2021: Metrics and baselines for unsupervised spoken language modeling. *arXiv preprint*.

Yoshihisa Ninokura, Hajime Mushiake, and Jun Tanji. 2003. Representation of the Temporal Order of Visual Objects in the Primate Lateral Prefrontal Cortex. *Journal of Neurophysiology*, 89(5):2868–2873.

Yoshihisa Ninokura, Hajime Mushiake, and Jun Tanji. 2004. Integration of Temporal Order and Object Information in the Monkey Lateral Prefrontal Cortex. *Journal of Neurophysiology*, 91(1):555–560.

Tanya Orlov, Volodya Yakovlev, Shaul Hochstein, and Ehud Zohary. 2000. Macaque monkeys categorize images by their ordinal number. *Nature*, 404(6773):77–80.

Tobias Overath, Rhodri Cusack, Sukhbinder Kumar, Katharina Von Kriegstein, Jason D Warren, Manon Grube, Robert P Carlyon, and Timothy D Griffiths. 2007. An Information Theoretic Characterisation of Auditory Encoding. *PLoS Biology*, 5(11):e288.

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, South Brisbane, Queensland, Australia. IEEE.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library.

Maritza Rivera-Gaxiola, Juan Silva-Pereyra, and Patricia K. Kuhl. 2005. Brain potentials to native and non-native speech contrasts in 7- and 11-month-old American infants. *Developmental Science*, 8(2):162–172.

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 and Natasha Z. Kirkham. 2018. Infant Statistical Learning. *Annual Review of Psychology*, 69(1):181–203.

Kapil Saini and Ajmer Singh. 2024. A content-based recommender system using stacked LSTM and an attention-based autoencoder. *Measurement: Sensors*, 31:100975.

Thomas Schatz, Francis Bach, and Emmanuel Dupoux. 2018. Evaluating automatic speech recognition systems as quantitative models of cross-lingual phonetic category perception. *The Journal of the Acoustical Society of America*, 143(5):EL372–EL378.

Thomas Schatz, Vijayaditya Peddinti, Francis Bach, Aren Jansen, Hynek Hermansky, and Emmanuel Dupoux. 2013. Evaluating speech features with the Minimal-Pair ABX task: Analysis of the classical MFC/PLP pipeline. In *Proceedings of INTERSPEECH 2013*, pages 1–5, Lyon, France.

Jan W. H. Schnupp, Thomas M. Hall, Rory F. Kokelaar, and Bashir Ahmed. 2006. Plasticity of Temporal Pattern Codes for Vocalization Stimuli in Primary Auditory Cortex. *The Journal of Neuroscience*, 26(18):4785–4795.

Anna Shestakova, Elvira Brattico, Alexei Soloviev, Vasily Klucharev, and Minna Huotilainen. 2004. Orderly cortical representation of vowel categories presented by multiple exemplars. *Brain Research. Cognitive Brain Research*, 21(3):342–350.

Benjamin Skerritt-Davis and Mounya Elhilali. 2018. Detecting change in stochastic sound sequences. *PLOS Computational Biology*, 14(5):e1006162.

M. Steinschneider. 2005. Intracortical Responses in Human and Monkey Primary Auditory Cortex Support a Temporal Processing Mechanism for Encoding of the Voice Onset Time Phonetic Parameter. *Cerebral Cortex*, 15(2):170–186.

Megha Sundara, Linda Polka, and Fred Genesee. 2006. Language-experience facilitates discrimination of /d-/ in monolingual and bilingual acquisition of English. *Cognition*, 100(2):369–388.

Frank Lihui Tan and Youngah Do. 2025. Attention-LSTM autoencoder simulation for phonotactic learning from raw audio input. *Linguistics Vanguard*.

Erik D. Thiessen and Jenny R. Saffran. 2003. When cues collide: Use of stress and statistical cues to word boundaries by 7- to 9-month-old infants. *Developmental Psychology*, 39(4):706–716.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the Practical Computational Power of Finite Precision RNNs for Language Recognition. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 740–745, Melbourne, Australia. Association for Computational Linguistics.

Da Ren Zhang, Zhi Hao Li, Xiang Chuan Chen, Zhao Xin Wang, Xiao Chu Zhang, Xiao Mei Meng, Sheng He, and Xiao Ping Hu. 2003. Functional comparison of primacy, middle and recency retrieval in human auditory short-term memory: an event-related fMRI study. *Cognitive Brain Research*, 16(1):91–98.

A Model Architecture Details

A.1 Encoder

The encoder was designed to extract latent information from the input, capturing essential features for further processing. To facilitate modeling, the original audio data was transformed into the Mel-Spectrogram features, which captures how different frequencies of sound change over time. The Mel-spectrogram features were structured in the shape of (B,T,F), where B is the batch size (set to 128), T is the sequence length, and F is 96 which corresponds to the number of Mel filter banks, defining the feature dimension for each frame. Zero-padding was applied to shorter sequences within the batch to maintain uniform sequence lengths. The input Mel-spectrogram was first passed through an FC layer, transforming the feature dimension from 96 to the target hidden dimension, reflecting differing memory capacities. The transformed features were then processed sequentially through five bidirectional LSTM layers, each integrating information from both past and future timesteps. This allowed the model to capture dependencies and patterns in the input data, enhancing its understanding of the sequential information. The output from the final LSTM layer was passed through another FC layer to integrate the concatenated bi-directional representation into a final output of the target dimension. This structure allowed the encoder to capture spectral as well as temporal patterns and generate a compact yet abstract representation of the input sequence, showing how the model learned the input data.

A.2 Hidden Representation

Each layer's output contained information at varying levels of abstraction, with deeper layers expected to capture more abstract and generalized features. The output of the encoder, i.e., the deepest hidden representation, encapsulated the knowledge learned by the encoder. This hidden representation formed the foundation for the decoder's reconstruction process, enabling the generation of output sequences. To further evaluate the influence of hidden dimensionality on the model's ability to learn phoneme contrast from contextual input, we experimented with six different hidden dimension settings: 4, 8, 16, 32, 48, and 64. Increasing the size of hidden dimensions aimed to mimic higher memory capacities.

A.3 Decoder and Cross-Attention

The aim of the decoder was to accurately reconstruct the input data using the learned information extracted by the encoder. It received the hidden representation the encoder generated as input and processed it autoregressively, meaning that each timestep's input was the output of the previous timestep. As shown in Figure 1, the decoder consists of an FC layer, five unidirectional LSTM layers, the cross-attention layer, and a final FC layer to generate the output. The decoding process began with a starting token, initialized to zeros, which served as the input for the first timestep. For each subsequent timestep, the decoder's LSTM layers incorporated historical information and made a prediction for reconstructing the next timestep. The scaled dot-product cross-attention mechanism

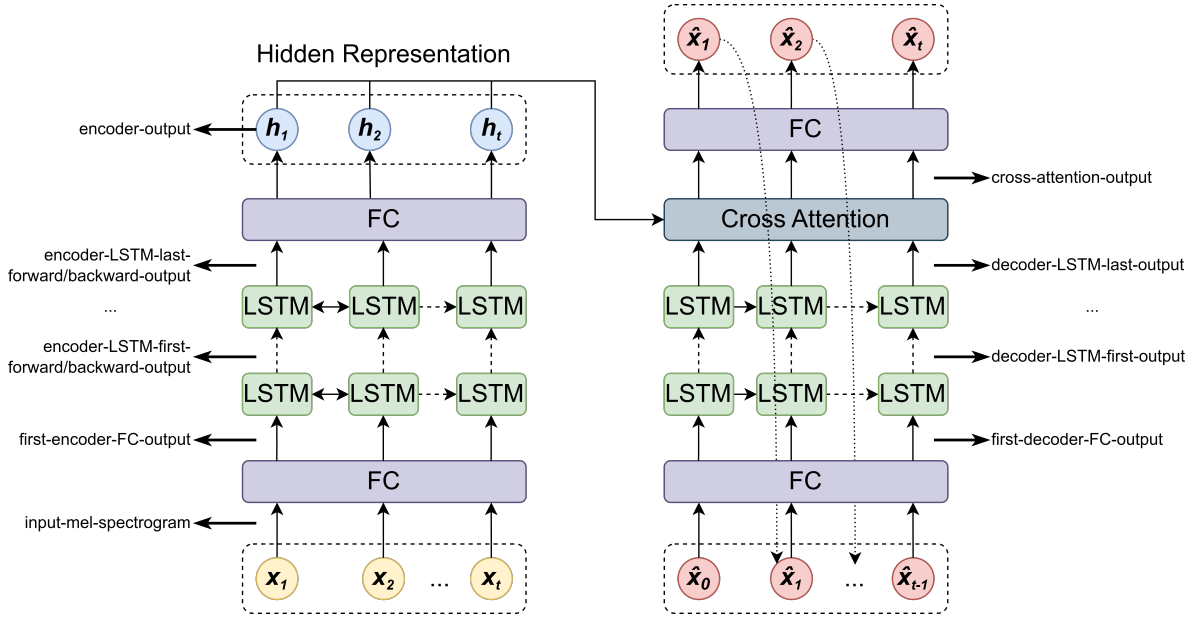


Figure 4: Illustration of layer output names.

(Vaswani et al., 2017) played a crucial role in bridging the encoder and decode. It allowed the decoder to query relevant information from the hidden representation. Specifically, the decoder's LSTM output served as the query, i.e., a representation of the current output that is used to retrieve relevant information, while the hidden representation acted as the keys and values, which helped the model to focus on important aspects of the input data. The cross-attention mechanism captured the similarity between the query and keys, guiding the decoder's reconstruction for the current timestep. After the attention layer, the data underwent a final FC transformation, shaping the output to match the input dimensions. This autoregressive output was then used as input for the next decoding timestep until the entire sequence was processed.

A.4 On Bi-directional LSTM

The use of bi-directional LSTMs was chosen to enhance the model's ability to integrate contextual information more effectively; this does not introduce additional non-humanlike characteristics, as the decoder's cross-attention mechanism will already incorporate global information.

A.5 Loss

This output was assessed against the input using the masked mean squared error (MSE) loss function (Paszke et al., 2019) to measure reconstruction accuracy.

B Dataset and Preprocessing Details

B.1 Concatenation

This approach allowed precise control over phoneme identity and phonotactic structure while minimizing systematic dependencies on particular lexical items or local contextual realizations. Segment selection was not constrained by positional or prosodic factors (e.g., word boundary, phrase position, or local coarticulatory context) but were instead sampled across a wide range of naturally occurring environments. This design was intended to avoid systematic coupling between specific positional contexts and acoustic realizations, thereby reducing the influence of strong but non-target position-acoustic correlations that are inherent in continuous speech.

B.2 Preprocessing

Mel-spectrograms were extracted from the audio recordings using the transforms.MelSpectrogram function from the torchaudio library (Hwang et al., 2023). This function transforms the input signal into a sequence of spectral samples through the application of filter banks for ease of processing. The n_fft and window length parameters were set to 512, and the hop length to 128. Using 96 Mel filter banks, the process produced windows of 32 ms with an 8 ms shift between consecutive windows. The resulting Mel-spectrogram had dimensions of (wave frame // 128, 96), where "wave frame" represents the total number of waveform frames. Finally,

mean and variance normalization was applied to the Mel-spectrogram to standardize input features and enhance consistency during processing.

C Evaluation Layer Outputs Visualization

Figure 4 displays the naming of layer output representations in more detail.

D Model Reconstruction Results Performances

We evaluated the model's performance on its primary training objective, namely reconstruction. Reconstruction quality visualization at the end of training can be found in the OSF link. Although the reconstruction lacked finer acoustic details, the model successfully captured the coarse energy distributions of sounds, which are crucial for their identification. The mean squared error (MSE) loss on the test set over 100 training epochs also demonstrated consistent improvement in reconstruction quality, reducing the MSE loss from 0.677 ($\sigma = 0.0459$) at epoch 0 to 0.179 ($\sigma = 0.0118$) by the end of training. This reconstruction performance confirms that the model was learning effectively, providing a reliable foundation for subsequent evaluations.

The Spanish Learner and Heritage Speaker Dependency Treebank

Valeria Pagliai **Sergio José Salazar Rodó**

Emiliana Pulido **Andres Gutierrez-Quintero**

Zoey Liu
University of Florida
liu.ying@ufl.edu

Introduction Data-driven analysis of syntactic development in second language (L2) acquisition requires learner corpora with (morpho)syntactic annotations (Wulff, 2017). Such datasets, however, remain scarce. While a few L2 corpora, focusing on English (e.g., the Learner English Treebank (TLE) (Berzak et al., 2016)), Italian (the learner Italian treebank (Di Nuovo et al., 2019)), or Korean (L2 Korean treebank (Sung and Shin, 2024)) provide syntactic annotations, they lack systematic documentation of learner deviations, i.e., linguistic patterns that reflect the learning and adoption of a language during the acquisition process.

To address these gaps, we present a manually curated L2 Spanish dataset (N = 49,247) based on the COWS-L2H corpus (Davidson et al., 2020). We adopted the Universal Dependencies (UD) (de Marneffe et al., 2021) framework, which offers clear annotation guidelines that are cross-linguistically adaptable and have been applied in previous L2 corpora (see above). Our annotations cover lemmatizations, part-of-speech (POS) tags, and syntactic dependencies. Compared to existing L2 corpora, our treebank: (1) captures learner deviation by considering instances of pro-drop, ungrammatical structures, and several dependency subrelations; (2) includes learners of different proficiency levels; (3) offers an additional dedicated annotation manual that can be straightforwardly employed and expanded by future research. Although there exists one Spanish L2 treebank (Pulido et al., 2025), it was annotated by a single annotator with 6,604 tokens only and contains no information of learner deviations. In contrast, our dataset constitutes the largest syntactic treebank for L2 Spanish with detailed annotations, which will be freely accessible.

Annotation process Four graduate students specializing in linguistics worked on the annotations. We first conducted a pilot phase in which we annotated 30 sentences randomly selected from the Spanish AnCora treebank (AnCora) (Taulé et al., 2008) from scratch. Inter-annotator agreement across the features of interest, including lemmas, POS tags, and syntactic dependencies, reached at least 90% on average. During this stage and throughout the initial phase of learner data annotations, one annotator constructed a detailed annotation manual providing instructions and examples on how to consistently address learner data.

We used the COWS-L2H corpus, which consists of 5,383 essays written by Spanish learners across different classes and instructional levels. The corpus includes lower, intermediate, advanced, and heritage speakers courses. We include at least one essay (N=168) from each of the 17 different courses, covering 165 unique student authors out of a total of 167, as some students contributed to multiple essays. After automatically parsing the learner essays with `spaCy`, annotators corrected parsing errors following the project's annotation guidelines. We added miscellaneous information about ungrammatical constructions, typos, and language identification. Finally, annotators also explicitly marked missing tokens in all sentences, including omissions of non-overt subjects (i.e., pro-drop).

Experiments Due to space limitation, we focus on the analysis of missing tokens and dependency parsing. Overall, the resulting treebank includes 2,415 missing tokens, the distribution of which appears to be influenced by both the POS tag of the omitted token and the proficiency level (Table 1): learners seem to drop pronouns (77.97%) and prepositions (9.65%) much more often than other POS tags. The frequency of missing tokens gradually decreases from lower to upper level classes, indicating potential improvement through the stages of acquisition.

With dependency parsing, we examined two data partitioning strategies coupled with different data representations. One split followed UD's guideline of including at least 10k tokens in the test set, and the other was based on the commonly used 8:1:1 ra-

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 127–128
July 3-4, 2026 ©2026 Association for Computational Linguistics

POS tag	Freq.
PRON	1,883
ADP	233
DET	99
PUNCT	50
Other	152

Level	Freq
Lower	1,075
Intermediate	905
Heritage	266
Upper	169

Table 1: Missing tokens and pro-drop across POS tags and class levels.

tio. For data representation, we experimented with prioritizing essays from upper-proficiency learners and from a combination of upper and intermediate learners in the training set. Additionally, we also explored excluding missing tokens from a given data split, as well as including our explicit annotations of such tokens. We used Label Attachment Scores (LAS) as the evaluation metric.

We carried out three training configurations, specifically, training parsers: (1) exclusively on AnCora, (2) solely on our manually annotated L2 data, and (3) on AnCora and subsequently fine-tuning on the L2 data. All models were trained with the MaChAmp toolkit (van der Goot et al., 2021) using the multilingual BERT-cased model (Devlin et al., 2019) and the default training parameters.

	AnCora	L2 treebank	Finetune AnCora
UD partition			
no empty tokens	0.858	0.866	0.875
with empty tokens	0.808	0.860	0.870
Regular partition			
no empty tokens	0.853	0.858	0.876
with empty tokens	0.814	0.865	0.877

Table 2: LAS scores from different partitioning strategies and training configurations, when training data contains both upper and intermediate learner data.

The different partitioning strategies and data representation yield reasonable LAS scores, ranging from 0.790 to 0.877. Table 2 presents results for when the training set consists of mostly upper and intermediate-level learner essays. With the UD partition, it seems that when excluding missing tokens from our annotated dataset, the parser derived from AnCora yields relatively good result (LAS=0.858); that said, we observe comparable performance when the parser was trained on the L2 data alone, which is approximately 13 time smaller compared to the AnCora training set. This demonstrates the effectiveness of our in-domain annotations. Finetuning Ancora with our L2 training data did not notably improve model performance, which potentially points to pronounced structural and distributional differences between Spanish L1 and L2 writing. Similar trends are present when missing tokens are explicitly labeled, across data splits and training configurations.

References

Yevgeni Berzak, Jessica Kenney, Carolyn Spadine, Jing Xian Wang, Lucia Lam, Keiko Sophie Mori, Sebastian Garza, and Boris Katz. 2016. Universal dependencies for learner English. In *Proceedings of the 54th Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 737–746.

Sam Davidson, Aaron Yamada, Paloma Fernandez Mira, Agustina Carando, Claudia H. Sanchez Gutierrez, and Kenji Sagae. 2020. Developing NLP tools with a new corpus of learner Spanish. In *Proceedings of the Twelfth LREC*.

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

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 NAACL:HLT, Volume 1 (Long and Short Papers)*.

Elisa Di Nuovo, Cristina Bosco, Alessandro Mazzei, Manuela Sanguinetti, and 1 others. 2019. Towards an Italian learner treebank in universal dependencies. In *CEUR Workshop Proceedings*, volume 2481, pages 1–6.

Emiliana Pulido, Robert Pugh, and Zoey Liu. 2025. I speak for the árboles: Developing a dependency treebank for Spanish L2 and heritage speakers. In *Proceedings of the 63rd Annual Meeting of the ACL (Volume 4: Student Research Workshop)*.

Hakyung Sung and Gyu-Ho Shin. 2024. Constructing a dependency treebank for second language learners of Korean. In *Proceedings of LREC-COLING 2024*.

Mariona Taulé, M. Antònia Martí, and Marta Recasens. 2008. AnCora: Multilevel annotated corpora for Catalan and Spanish. In *Proceedings of the Sixth International Conference on LREC*.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the EACL: System Demonstrations*.

Stefanie Wulff. 2017. What learner corpus research can contribute to multilingualism research. *International Journal of Bilingualism*, 21(6):734–753.

Omnivorous Agreement, like Uyghur Backness Harmony, is a Challenge for Tier-Based Strict Locality

Allison Verbil
Department of Linguistics
University of California, Los Angeles
averbil@ucla.edu

Tim Hunter
Department of Linguistics
University of California, Los Angeles
timhunter@ucla.edu

Abstract

A well-known exception to the characterization that phonological patterns belong to the subregular class of TSL dependencies is found in Uyghur backness harmony (Mayer and Major, 2018). At the same time, a recent line of work has argued that many long-distance syntactic phenomena are subsumed by the TSL class, revealing an interesting parallel between phonology and syntax. We show that a certain omnivorous syntactic agreement pattern, namely Mundari object agreement (Murugesan et al., 2025), poses the same challenge to TSL as Uyghur backness harmony.

1 Introduction

It is generally accepted that phonological processes are at most regular stringsets and string-to-string mappings (Johnson, 1972; Kaplan and Kay, 1994), and can be computed by regular grammars and finite-state automata (Heinz, 2018). This regular class within the Chomsky Hierarchy is, however, argued to not be sufficiently restrictive in characterizing possible phonological patterns, and it has been claimed that a stronger characterization — that phonological stringsets are tier-based strictly local languages, which are subregular — holds (Heinz, 2018). This hypothesis claims that valid segmental phonological strings can be captured via constraints on substrings, except that these substrings may belong to a tier which contains only the segments relevant to a particular process (Heinz et al., 2011). Mayer and Major (2018) show, however, that a segmental process in Uyghur cannot be captured by a TSL language due to the specific behavior of potential harmony triggers in Uyghur.

At the same time, recent work has argued that something similar holds in the domain of syntax, namely that many agreement patterns fall within the class of TSL languages when construed as constraints on strings representing "spines" of a tree (Hanson, 2025). This paper will argue that a certain *omnivorous agreement* pattern (Nevins, 2011), namely object agreement in Mundari (Murugesan et al., 2025) poses a challenge for this approach to agreement that is analogous to the challenge posed by Uyghur in the phonological domain. The critical property of this omnivorous agreement pattern is that second person DPs, which rank in the middle of a three-level person scale, are sometimes, but not always, relevant for determining the form of the object marker; we argue that this is analogous to the way certain dorsal consonants in Uyghur are sometimes, but not always, relevant to the backness harmony. We will show that, because of this formal similarity, a natural extension of TSL that increases the complexity of the tier projection function to be sensitive to symbols recently projected onto the tier (Graf and Mayer, 2018; De Santo and Graf, 2019; Graf and Shafiei, 2019) captures both the Uyghur and Mundari patterns in a unified way.

The paper is structured as follows. §2 provides a background on TSL grammars in phonology and syntax, §3 summarizes Mayer and Major (2018)'s claims about Uyghur, and §4 shows Mundari omnivorous agreement behaves analogously to Uyghur. §5 presents potential objections, §6 introduces O-TSL grammars for Uyghur and Mundari, and §7 concludes.

2 Background on TSL

2.1 The TSL class

Σ represents a finite set of symbols referred to as an *alphabet*, and Σ^* denotes the set of all strings over Σ of any finite length. The empty string is ϵ. $\rtimes$ and $\ltimes$ will be edge markers, which denote the left and right edge of a word, respectively, and are not part of Σ. A string u is a substring of a string w iff $w = xuy$ for some strings $x, y \in \Sigma^*$. The k-factors of a string are its k-length substrings. The function $f_k(w)$ returns the set of k-factors of w.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 129–137
July 3-4, 2026 ©2026 Association for Computational Linguistics

Informally, a strictly k-local grammar is a finite set of forbidden substrings of length k, and the strictly local language generated by that grammar is the set of all strings that do not contain any of those forbidden substrings. More formally, we can say that a strictly k-local grammar has a finite set of k-factors $G \subseteq (\{\rtimes, \ltimes\} \cup \Sigma)^k$ which describes forbidden, or illicit, substrings. The language generated by a strictly k-local grammar G contains a string $w \in \Sigma^*$ iff $\rtimes w \ltimes$ contains no illicit substrings, i.e. $w \in L(G)$ iff $f_k(\rtimes w \ltimes) \cap G = \emptyset$. A language L is strictly local iff, for some $k \in \mathbb{N}$, there is a strictly k-local grammar that generates L.

Tier-based strictly local languages, as a class, properly contain the strictly-local languages. Informally, tier-based strictly local grammars differ from strictly-local grammars in that only symbols projected onto some tier are checked for illicit substrings. Formally, we can define a tier $\tau \subseteq \Sigma$, and a projection function $E_\tau(\sigma_1 \dots \sigma_n) = u_1 \dots u_n$, where $u_i = \sigma_i$ iff $\sigma_i \in \tau$ and $u_i = \epsilon$ otherwise. This removes symbols from the string that are not in τ. The k-factors for a k-TSL grammar are taken from $(\{\rtimes, \ltimes\} \cup \tau)^k$. The language generated by a k-TSL grammar G contains a string $w \in \Sigma^*$ iff the tier-projection of w, enriched with edge markers, contains no illicit substrings, i.e. $w \in L(G)$ iff $f_k(\rtimes E_\tau(w) \ltimes) \cap G = \emptyset$. A language L is TSL iff, for some $k \in \mathbb{N}$, there is a k-TSL grammar that generates L.

2.2 TSL patterns in phonology

As the k-factors in a TSL grammar are substrings on a tier, we can account for certain kinds of long-distance phonotactic dependencies like vowel and consonant harmony with blocking (Heinz et al., 2011). Consider a language like Georgian, which exhibits long-distance liquid dissimilation, where the suffix /-uri/ surfaces as [-uli] when preceded by an [r] anywhere in the stem (Fallon, 1993):

(1) a. /p'olon-uri/ → [p'o**l**on-u**r**i] 'Polish'
b. /ungr-uri/ → [ung**r**-u**l**i] 'Hungarian'
c. /gramat'ik'a-uri/ → [g**r**amat'ik'-u**l**i] 'grammatical'

However, dissimilation is blocked when /-uri/ is preceded by an intervening [l], as in [avst'**ral**-u**r**i] 'Australian' and [ka**rtl**-u**r**i] 'Kartvelian'. If our tier alphabet contains only the symbols τ = {r,l} we can forbid adjacent identical liquids by banning the 2-factors [rr] and [ll] on the tier, which will rule out any strings that do not obey dissimilation. For example, the tier projection of the illicit word *[ungr-uri] is [rr], which (non-properly) contains the banned substring [rr]. Cases where blocking occurs are straightforward to handle with this TSL-2 grammar; the tier projection of the licit string [kartl-uri] is [rlr], and therefore does not have two adjacent [r] segments, correctly predicting that [kartl-uri] is licit. The vast majority of observed phonological patterns that have been analyzed with TSL grammars have, like this example, used $k = 2$ (Jardine and Heinz, 2016).

2.3 TSL patterns in syntax

More recently there have been suggestions that TSL stringsets, also with $k = 2$, might play an interesting role in characterizing syntactic patterns (Graf and Shafiei, 2019; Hanson, 2025). The relevant syntactic objects are trees, as usual, but the idea in this work is that syntactic constraints might be expressible as conditions on strings that correspond to "spines" of the tree. These are known as c-strings.

We assume the relevant trees are headed binary trees[1] with lexical items at the leaves, and we consider *only pre-movement positions*. Any such tree is the maximal projection of some lexical item. The c-string of maximal projection XP is the string of lexical items comprised of: (i) the c-string of XP's complement (if it has one), followed by (ii) the heads of each of XP's specifiers (if it has any), in deepest-first order, followed by (iii) the head of XP. This is illustrated in Figure 1.

Following Hanson (2025), the simple English agreement pattern in (2) can be expressed in terms of the c-strings extracted from trees of the form shown in Figure 2.

(2) a. This dog chase**s** these cats
b. * This dog chase∅ these cats
c. * These cats chase**s** this dog
d. These cats chase∅ this dog

Taking into account (only) the base position of the subject, and abstracting away from whatever post-syntactic process leads to the T head surfacing in a position attached to the verb, we arrive at the four c-strings shown in (3).

(3) a. cats these chase this v T_{-s}

[1] By this we mean a binary tree where each internal node is designated as a projection of exactly one of its daughters. This is sufficient for the notions of maximal projection, head, complement and specifier to be determined in standard ways.

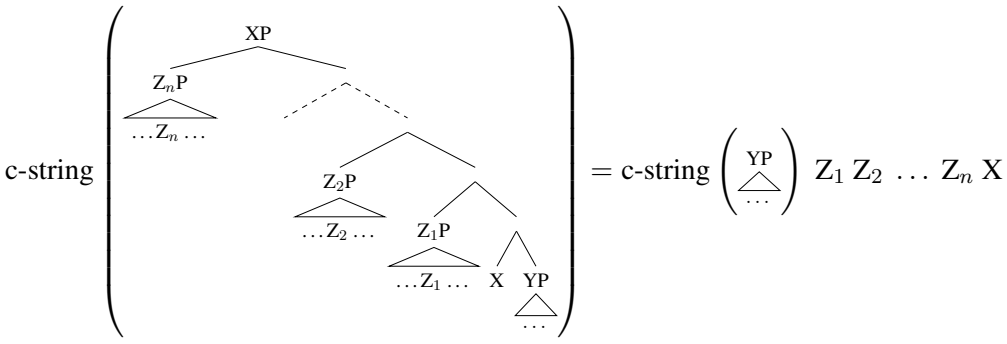


Figure 1: Schematic illustration of the definition of a tree's c-string. Notice that the c-string of XP includes the full c-string of its complement YP (which will end with the head Y), but only the heads Z_i of its specifiers.

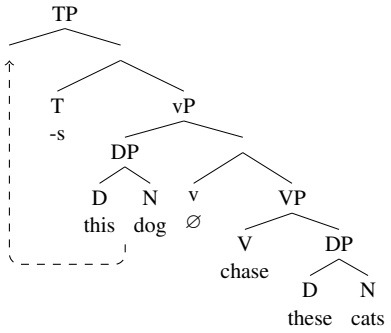


Figure 2: Tree structure for sentence (2a)

b. cats these chase this v $T_\varnothing$
c. dog this chase these v $T_{\text{-s}}$
d. dog this chase these v $T_\varnothing$

It will be useful to write these strings in a form that more clearly indicates the relevant linguistic properties. For this we adopt the strategy, discussed by Aksënova et al. (2016), of "relabeling" surface forms, in order to deal with strings of symbols such as D_{sg} and D_{pl} to collapse irrelevant distinctions (e.g. between 'these' and 'those'). Then instead of (3) we have (4).

(4) a. N D_{pl} V D_{sg} v $T_{\text{-s}}$
b. N D_{pl} V D_{sg} v $T_\varnothing$
c. N D_{sg} V D_{pl} v $T_{\text{-s}}$
d. N D_{sg} V D_{pl} v $T_\varnothing$

Picking out the c-strings corresponding to the acceptable sentences can be achieved by a TSL-2 grammar: specifically, if we project only D and finite T heads (i.e. $\tau = \{D_{sg}, D_{pl}, T_{\text{-s}}, T_\varnothing\}$), then the relevant requirement can be stated as a ban on the tier bigrams 'D_{sg} $T_\varnothing$' and 'D_{pl} $T_{\text{-s}}$'.

(5) a. N D_{pl} V D_{sg} v $T_{\text{-s}}$ $\xrightarrow{E_\tau}$ D_{pl} D_{sg} $T_{\text{-s}}$ ✓
b. N D_{pl} V D_{sg} v $T_\varnothing$ $\xrightarrow{E_\tau}$ D_{pl} D_{sg} $T_\varnothing$ ✗
c. N D_{sg} V D_{pl} v $T_{\text{-s}}$ $\xrightarrow{E_\tau}$ D_{sg} D_{pl} $T_{\text{-s}}$ ✗
d. N D_{sg} V D_{pl} v $T_\varnothing$ $\xrightarrow{E_\tau}$ D_{sg} D_{pl} $T_\varnothing$ ✓

The use of bigrams here matches the length of k-factors that have been used in phonology.

Although in these examples the T head is at a fixed distance from the D head it depends on (only v intervenes), the use of the tier allows for the same constraint to generalize to agreement in existentials; c-strings exclude the expletive 'there' for the same reason that they ignore post-movement positions.

(6) a. There seems to be a cat here
$\xrightarrow{E_\tau}$ D_{sg} $T_{\text{-s}}$ ✓
b. * There seem$\varnothing$ to be a cat here
$\xrightarrow{E_\tau}$ D_{sg} $T_\varnothing$ ✗
c. * There seems to be two cats here
$\xrightarrow{E_\tau}$ D_{pl} $T_{\text{-s}}$ ✗
d. There seem$\varnothing$ to be two cats here
$\xrightarrow{E_\tau}$ D_{pl} $T_\varnothing$ ✓

By also projecting C heads to the tier, and banning bigrams where a C head is followed by $\varnothing_T$, this TSL grammar over c-strings can capture the appropriate pattern for finite complements of verbs like 'seem' and for sentential subjects.

(7) It seems that these cats sit$\varnothing$
w = V D_{pl} v $T_\varnothing$ C V v $T_{\text{-s}}$
$E_\tau(w)$ = D_{pl} $T_\varnothing$ C $T_{\text{-s}}$ ✓

(8) That these cats sit$\varnothing$ surprises us
w_1 = D_{pl} V C v $T_{\text{-s}}$
w_2 = V D_{pl} v $T_\varnothing$
$E_\tau(w_1)$ = D_{pl} C $T_{\text{-s}}$ ✓
$E_\tau(w_2)$ = D_{pl} $T_\varnothing$ ✓

Note that for (8) there are two non-trivial c-strings: w_1 for the main clause's spine, on whose tier projection the C head separates the plural object D_{pl} ('us') from the T head, and w_2 for the sentential subject itself, where the standard agreement with 'these cats' is enforced as usual.

In sum, the TSL-2 grammar that covers the basic English agreement facts discussed here can be formalized as follows:

(9) $$\tau = \{D_{sg}, D_{pl}, C, T_{\text{-s}}, T_{\varnothing}\}$$
$$G = \{D_{sg}\ T_{\varnothing},\ D_{pl}\ T_{\text{-s}},\ C\ T_{\varnothing}\}$$

This TSL-2 grammar follows, in all essential respects, the proposal in Hanson (2025), although we have focused on characterizing the distributions of, and dependencies among, elements that are distinguished in relatively pre-theoretic terms (D_{sg}, $T_{\text{-s}}$, etc.), whereas Hanson presents grammars that regulate the distribution of diacritics ($[+\phi]$ and $[-\phi]$) that are assumed to drive a feature-copying agreement operation; see Hanson (2025, pp.62-63), for discussion of this choice point. We have opted for the former approach here because it more straightforwardly parallels the way vowel harmony patterns are typically described.

One more inconsequential difference between our presentation here and Hanson's is that Hanson writes c-strings in the reverse order, with heads at the beginning and embedded complement c-strings at the end, whereas we have followed the ordering introduced originally in Graf and Shafiei (2019); again, the reason for our choice here is to bring out the parallels with vowel harmony more clearly.

While our focus in this section has been on c-strings as a method of enforcing agreement patterns via TSL constraints, following Hanson (2025), Graf and Shafiei (2019) apply the c-string approach to other syntactic phenomena, and make use of a slightly more powerful formalism: IO-TSL grammars. We return to the relationship between our work and that of Graf and Shafiei (2019) in §6.2.

3 The challenge from Uyghur

One of the most well-known challenges to the generalization that harmony patterns in phonology are TSL comes from vowel harmony in Uyghur. Uyghur exhibits a backness harmony pattern which requires suffixes to agree in backness with certain vowels and consonants in the stem (Mayer and Major, 2018). The relevant segments that participate in harmony are shown in Table 1.

	Harmonizing vowels	**Harmonizing consonants**
Front	y, ø, æ	k, g
Back	u, o, a	q, ʁ

Table 1: Segments which participate in Uyghur backness harmony. The vowels [i, e] are transparent non-participatory vowels.

	Form	**Gloss**	**Harmony type**
a.	aʁinæ-**dæ**	friend-LOC	closest front vowel
b.	qoichi-**da**	shepherd-LOC	closest back vowel
c.	gezit-**tæ**	newspaper-LOC	closest front dorsal
d.	qirʁiz-**da**	Kyrgyz-LOC	closest back dorsal
e.	rak-**ta**	shrimp-LOC	closest back vowel, across front dorsal
f.	mæʃq-**tæ**	exercise-LOC	closest front vowel, across back dorsal
g.	it-**ta**	dog-LOC	no trigger, arbitrarily back
h.	biz-**dæ**	we-LOC	no trigger, arbitrarily front

Table 2: Examples of Uyghur backness harmony. Alternations of /-DA/ are bolded, and the harmony triggers are highlighted in blue. From Mayer and Major (2018).

The process for determining the backness value of a particular suffix can be described as follows:

(10) a. If there is a harmonizing vowel in the stem, the suffix should match the backness of the final harmonizing vowel: [-da] if back, [-dæ] if front.

b. If there's no harmonizing vowel in the stem, the suffix should match the backness of the final harmonizing consonant: [-da] if uvular, [-dæ] if velar.

c. If there's no harmonizing vowel or harmonizing dorsal in the stem, the suffix can surface as either [-da] or [-dæ].

Representative examples of the above process can be seen in Table 2, from Mayer and Major (2018). The suffix depicted is /-DA/, the locative case marker. Note that there is a voicing alternation in the suffix that is orthogonal to the present pattern; the description in (10) abstracts away from this alternation. In case (10c), it is lexically-determined, for any given stem, which of the two forms appears, but we treat this as distinct from the harmony requirement itself (i.e. while [it-**tæ**] is not actually a valid surface form, we do not take it to be in violation of the harmony pattern that we are modeling).

The essential takeaway is that harmonizing vowels will always take precedence over dorsal consonants as harmony triggers (rows e–f), and harmo-

nizing consonants only serve as triggers when there is insufficient backness information available from the vowels (rows c–d). Mayer and Major's claim is that it is this exact nature — where both vowels and consonants can be harmony triggers, but consonants can only be considered in the absence of vowels — that is problematic for TSL.

To describe this pattern formally, we'll relabel all harmonizing front and back vowels in Table 2 as V_f and V_b respectively, and similarly relabel harmonizing consonants as C_f and C_b. Forms of the suffix with a front vowel are relabeled S_f, and those with a back vowel S_b. The pattern is at most regular, since it can be characterized by the regular expression $r_1 \mid r_2 \mid r_3 \mid r_4 \mid r_5$, where $V = V_f \mid V_b$, $C = C_f \mid C_b$, and:

(11)
$$\begin{aligned} r_1 &= (V \mid C \mid X)^* \, V_f \, (V_f \mid C \mid X)^* \, S_f \\ r_2 &= (V \mid C \mid X)^* \, V_b \, (V_b \mid C \mid X)^* \, S_b \\ r_3 &= (C \mid X)^* \, C_f \, (C_f \mid X)^* \, S_f \\ r_4 &= (C \mid X)^* \, C_b \, (C_b \mid X)^* \, S_b \\ r_5 &= X^* \, (S_f \mid S_b) \\ X &= \overline{(V \mid C \mid S_f \mid S_b)} \end{aligned}$$

Mayer and Major note that if we set aside cases where the suffix harmonizes with a consonant (i.e. consider only $r_1 \mid r_2 \mid r_5$), a TSL-2 grammar where we project all and only suffixes and harmonizing vowels would suffice.

(12)
$$\begin{aligned} \tau &= \{V_f, V_b, S_f, S_b\} \\ G &= \{V_f S_b, \; V_b S_f\} \end{aligned}$$

This implements a ban on vowel-suffix sequences on the tier that mismatch in backness, preventing forms like *[aʁinæ-da] where the suffix does not harmonize with the stem-final vowel.

The challenge comes from how to incorporate the role of the harmonizing consonants. If these harmonizing consonants are omitted from the tier, then they will never have any non-local effect at all on the predicted suffix forms, which clearly cannot work. If they are included in the tier, then this will allow correctly for cases like [gezit-**tæ**] where the harmonizing consonant is the only harmonizing element in the stem (rows c–d); but crucially, this will lead to incorrect results in cases where a harmonizing consonant intervenes between a harmonizing vowel and a suffix (rows e–f). And since there is no bound on the number of such intervening consonants, any k-TSL grammar will be unable to make the necessary distinctions between cases that lead to the tier projections in (13), because of the formal property that Mayer and Major refer to known as suffix substitution closure; intuitively, the critical vowel is too far away from the suffix for the k-factors to enforce the desired result.

(13)
$$\begin{array}{ll} V_b \, C_f^{k-1} \, S_f & V_f \, C_f^{k-1} \, S_f \\ V_b \, C_b^{k-1} \, S_b & V_f \, C_b^{k-1} \, S_b \end{array}$$

Put differently, there's no one-size-fits-all answer to the question of whether C_f and C_b should be projected onto the tier. For this pattern to be enforced by strictly-local constraints on a tier, the projection function would need to project C_f and C_b only in the absence of V_f or V_b.

4 The analogous challenge from Mundari

As discussed in §1, a recent line of work has argued that many long-distance syntactic dependencies, including agreement, are subsumed by the TSL-2 class (Hanson, 2025, among others). Here, we argue that *omnivorous agreement* presents a challenge to TSL that is analogous to the above problem presented by Uyghur. *Omnivorous agreement* (Nevins, 2011) refers to a pattern where an agreement marker tracks the highest-ranking argument along some given hierarchy, regardless of the relative positions of the arguments.

Mundari (Austroasiatic, Murugesan et al., 2025) exhibits omnivorous agreement in ditransitive constructions, where both the direct and indirect object compete for a single agreement slot[2]. The language shows complex interactions between person, number, and animacy hierarchies, but we restrict attention here to cases where both objects are singular and animate; in these scenarios, object agreement follows a 1>2>3 person hierarchy. For example, consider the interaction between first and second person DPs in (14):

(14) a. hon-ko aiɲ ke am ke-ko
children-PL 1SG EMP 2SG EMP-3PL.SM
ɛm-a-iɲ-ta-n-a
give-APPL-1SG.OM-PROG-ITR-IND
"Children are giving me to you."
1SG DO & 2SG IO

b. hon-ko am ke aiɲ ke-ko
children-PL 2SG EMP 1SG EMP-3PL.SM
ɛm-a-iɲ-ta-n-a
give-APPL-1SG.OM-PROG-ITR-IND

[2] Murugesan et al. (2025) note that this pattern is clitic doubling and not "true" inflectional agreement, but this is orthogonal to our argument.

"Children are giving you to me."
2SG DO & 1SG IO

Here, first person DPs are tracked instead of second person, regardless of whether the first person DP is the direct object or the indirect object. When no first person DP is present, second person DPs are tracked over third person DPs regardless of object status, as in (15):

(15) a. hon-ko am ke Ravi ke-ko
children-PL 2SG EMP Ravi EMP-3PL.SM
ɛm-a-m-ta-n-a
give-APPL-2SG.OM-PROG-ITR-IND
"Children are giving you to Ravi."
2SG DO & 3SG IO

b. hon-ko Ravi ke am ke-ko
children-PL Ravi EMP 2SG EMP-3PL.SM
ɛm-a-m-ta-n-a
give-APPL-2SG.OM-PROG-ITR-IND
"Children are giving Ravi to you."
3SG DO & 2SG IO

A combination of first and third person will yield the same agreement marker '-iŋ' as in (14); in a case with two third person (singular, animate) objects (i.e. no first or second person DPs are present), the distinct marker '-i' surfaces. In sum, the pattern can be described as follows:

(16) a. If there is a first person DP in the agreement domain, the object marker is '-iŋ'.

b. If there's no first person DP in the agreement domain, but there is a second person DP, the object marker is '-m'.

c. If there's no first person or second person DP in the agreement domain, the object marker is '-i'.

We take the underlying (pre-movement) structure of Mundari's ditransitives to be as shown in Figure 3, with the Appl head surfacing as the object marker (Murugesan et al., 2025). As before, we work with a relabeling of syntactic elements that focuses attention on the critical distinctions: we use Ap_A, Ap_B and Ap_C for applicative heads that surface as the '-iŋ', '-m' and '-i' object markers respectively, and D_1, D_2 and D_3 for first, second and third person D heads. Given the tree structure in Figure 3, where the indirect object c-commands the direct object, the relevant c-strings for (14) and (15) will be as follows. (Note that the subject is third person throughout these examples; we focus only on the effect of varying the objects.)

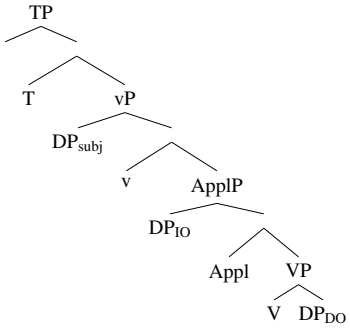


Figure 3

(17) a. (14a), 1SG DO & 2SG IO
c-string: D_1 V D_2 Ap_A D_3 v T

b. (14b), 2SG DO & 1SG IO
c-string: D_2 V D_1 Ap_A D_3 v T

(18) a. (15a), 2SG DO & 3SG IO
c-string: D_2 V D_3 Ap_B D_3 v T

b. (15b), 3SG DO & 2SG IO
c-string: D_3 V D_2 Ap_B D_3 v T

(19) a. 3SG DO & 3SG IO
c-string: D_3 V D_3 Ap_C D_3 v T

b. 3SG DO & 3SG IO
c-string: D_3 V D_3 Ap_C D_3 v T

The pattern described in (16), construed as a constraint on c-strings, can therefore be characterized by the regular expression $r_1 \mid r_2 \mid r_3$, where:

(20)
$$r_1 = (\text{D}_1 \mid \text{D}_2 \mid \text{X})^* \, \text{D}_1 \, (\text{D}_1 \mid \text{D}_2 \mid \text{X})^* \, \text{Ap}_\text{A} \, \text{X}^*$$
$$r_2 = (\text{D}_2 \mid \text{X})^* \, \text{D}_2 \, (\text{D}_2 \mid \text{X})^* \, \text{Ap}_\text{B} \, \text{X}^*$$
$$r_3 = \text{X}^* \, \text{Ap}_\text{C} \, \text{X}^*$$
$$\text{X} = \overline{(\text{D}_1 \mid \text{D}_2 \mid \text{Ap}_\text{A} \mid \text{Ap}_\text{B} \mid \text{Ap}_\text{C})}$$

Notice the parallels with (11), with D_1 playing a role analogous to vowels, and D_2 to consonants.

Like Uyghur, we will show that the problematic nature of this paradigm results from second person DPs only being considered as agreement goals in the absence of first person DPs; this is analogous to the way dorsal consonants in Uyghur are harmony triggers only in the absence of harmonizing vowels.

If we were not concerned with agreement triggered by second person DPs (i.e. if we consider only $r_1 \mid r_3$), then an appropriate TSL grammar could be constructed along the same lines as (12): project only first person DPs and object markers, and ensure that the choice between Ap_A and Ap_C

tracks the presence or absence of D_1:

(21) $\tau = \{D_1, Ap_A, Ap_C\}$
$G = \{D_1 Ap_C,\ \rtimes Ap_A\}$

The challenge is in incorporating sensitivity to the presence of second person DPs. If we omit second person DPs from the tier, then it will be impossible to regulate the proper choice between Ap_B and Ap_C in (18) and (19). If we include second person DPs on the tier, then we can correctly account for (18) and (19), but the second person indirect object will intervene between the first person direct object and the applicative head in (17a)/(14a). As in the Uyghur case then, no choice of tier symbols will allow a TSL grammar to enforce the observed pattern.

More generally, any omnivorous agreement system which makes use of a three-level scale to determine the surface form of an agreement marker cannot be captured by a TSL grammar. The heart of the issue is that the middle level of such a scale is sometimes, but not always, relevant; neither always projecting it to the tier, nor always omitting it from the tier, will suffice. According to the typology reported by Murugesan et al. (2025), such patterns are relatively rare even among omnivorous agreement systems, although the same 1>2>3 person scale as Mundari has also been observed in Alutor (Chukotkan). Omnivorous person agreement more commonly uses a two-level scale, such as in Kichean, Chuckchi (Chukotkan), and Eastern Armenian, where first and second person are incomparable ({1,2}>3); these patterns will therefore pose no problem for TSL grammars. In the domain of omnivorous number agreement, Murugesan et al. (2025) note that two-level scales are again more common (e.g. PL > SG in Georgian, Kichean, Mara (Tibeto-Burman), or SG > PL in Ketama Berber), but TSL-incompatible three-level scales have also been observed, for example in Hayu (Sino-Tibetan) and Onondaga (Northern Iroquoian). In fact, Mundari itself also uses a three-level SG > PL > DU number scale in object agreement when animacy and person values are held constant.

5 Differences between Uyghur backness harmony and omnivorous agreement

5.1 Multiple "domains" in a single c-string

In the formal language that Mayer and Major (2018) argued is not TSL, as reviewed in §3 above, each string corresponds to one inflected Uyghur word, consisting of a stem plus an appropriate suffix. We have suggested that the way in which that suffix's surface form depends on its phonological context is analogous to the way in which the object agreement marker in Mundari depends on its syntactic context. While the analysis in §4 restricted attention to monoclausal sentences, which contain at most one object agreement marker, c-strings representing full Mundari sentences may contain multiple clauses; and a complete formal grammar for these c-strings will therefore allow strings that contain multiple object agreement markers, each with a surface form that is dependent on the DPs in the corresponding clause. So one difference between a grammar for Uyghur word forms and a grammar for Mundari c-strings will be that each string generated by the former consists of only one "harmony domain", whereas a string generated by the latter might consist of multiple distinct "agreement domains" concatenated. This difference is orthogonal, however, to our key point which is that the patterns observed inside each domain are analogous, and beyond the capacity of TSL grammars.

5.2 Unboundedness of DPs in an agreement domain

A more significant potential objection to our main claim concerns the question of how many DPs can occur in a single agreement domain (as opposed to the question of how many agreement domains can occur in a single string). In (20) we characterized the pattern in a way that allows for unboundedly many D heads in the domain that an object marker's surface form depends on, and one might question this given that all the example sentences in §4 included just two DPs in the relevant domain (a direct object and an indirect object). If the number of DPs in a domain were limited to two, then the pattern would be TSL: it would suffice to project *all* D heads onto the agreement tier, since there would be a limit on how far away the highest-ranking D head could be from the object marker. This would be crucially different from the situation in Uyghur, where it is natural to assume that unboundedly many dorsal consonants might intervene between a critical harmonizing vowel and the suffix (recall (13)).

The formulation we gave in (20) is based on an assumption that, while it may indeed be impossible to construct an acceptable Mundari sentence with more than two DPs in a single object-agreement domain, this reflects a property of some part of

the language's grammar that is distinct from its agreement mechanisms. We conjecture that the explanation of this boundedness of DPs will come from theories of thematic structure or argument licensing, and that it would therefore be redundant to build into our theories of agreement a limit on how many DPs might need to be considered. This separation of concerns is consistent with the way the mechanisms responsible for agreement patterns are described in the syntactic literature (e.g. Chomsky, 2001; Béjar, 2003): in such theories, a "probe" typically searches its c-command domain for an appropriate "goal" in a manner that does not involve keeping track of how much structure has been searched so far.

Having said that, even if one rejects our assumed separation of the number of DPs from the mechanisms of agreement, there is still something significant about the Mundari object agreement pattern. The overwhelming majority of TSL accounts of linguistic patterns (in both phonology and syntax) fall within the particularly restrictive TSL-2 class, i.e. they are stated in terms of length-two substrings on the relevant tier (see e.g. Hanson, 2025, pp.87–88; Jardine and Heinz, 2016, p.88). If the number of relevant DPs for Mundari object agreement were limited to two, then a TSL grammar could achieve the desired result by projecting all D heads onto the tier, but this would be a TSL-3 grammar, since the k-factors would include both objects and the agreement marker. The strong non-TSL claim would hold only for Uyghur vowel harmony, but the weaker (and still linguistically significant) non-TSL-2 claim would still hold in a unified way for Uyghur vowel harmony and Mundari object agreement.

6 Omnivorous agreement is O-TSL

6.1 O-TSL

Mayer and Major (2018, p.73) point out that Uyghur backness harmony can be formulated as a strictly local constraint on a tier if the projection function, rather than simply erasing certain symbols, is an output-sensitive strictly local map (Chandlee, 2014). This puts the harmony pattern itself in the class of O-TSL languages. To sharpen the parallels between Uyghur backness harmony and omnivorous agreement, this section will show how O-TSL grammars can capture both patterns.

Formally, a (j, k)-O-TSL grammar is like a k-TSL grammar, except that the tier projection function can take into account the most recent $j-1$ symbols already projected to the tier when deciding whether to project a particular surface symbol.[3] So to define the tier projection function, we need to specify not simply a subset τ of Σ, but rather a set of (σ, t) pairs, called j-contexts, where $\sigma \in \Sigma$ and $t \in (\{\rtimes\} \cup \Sigma)^{j-1}$. The intended interpretation of a j-context (σ, t) is that an occurrence of σ in the surface string is to be projected onto the tier iff the tier constructed so far ends in t.

As Mayer and Major point out, the Uyghur harmony pattern becomes strictly local on the relevant tier representations if we specify that harmonizing consonants are projected only if the most recent symbol on the tier is not a harmonizing vowel. This ensures that the suffix does not end up adjacent to an irrelevant intervening dorsal consonant in cases like [rak-**ta**] and [mæʃq-**tæ**]. The (2,2)-O-TSL grammar that achieves this is as follows.

$$C = \{(\sigma, t) \mid \sigma \in \{\mathrm{V_f}, \mathrm{V_b}, \mathrm{S_f}, \mathrm{S_b}\}, t \in \Sigma\} \cup \{(\sigma, t) \mid \sigma \in \{\mathrm{C_f}, \mathrm{C_b}\}, t \in \Sigma - \{\mathrm{V_f}, \mathrm{V_b}\}\}$$
$$G = \{\mathrm{V_f S_b}, \mathrm{V_b S_f}, \mathrm{C_f S_b}, \mathrm{C_b S_f}\}$$

An exactly analogous strategy works for the omnivorous person agreement pattern that we described above: we project $\mathrm{D_2}$ only if the most recent symbol on the tier is not $\mathrm{D_1}$. This prevents a $\mathrm{D_2}$ from ending up adjacent on the tier to the agreement marker in a scenario where a $\mathrm{D_1}$ is present.

$$C = \{(\sigma, t) \mid \sigma \in \{\mathrm{D_1}, \mathrm{Ap_A}, \mathrm{Ap_B}, \mathrm{Ap_C}\}, t \in \Sigma\} \cup \{(\mathrm{D_2}, t) \mid t \in \Sigma - \{\mathrm{D_1}\}\}$$
$$G = \{\mathrm{D_1 Ap_B},\ \mathrm{D_1 Ap_C},\ \mathrm{D_2 Ap_A},\ \mathrm{D_2 Ap_C}\}$$

6.2 TSL, O-TSL, and IO-TSL

This analysis of omnivorous agreement is significant given the conjecture in Hanson (2025) that syntactic agreement will fall within TSL. However, as mentioned above in §2.3, Graf and Shafiei (2019) used tier-based constraints on c-strings to model syntactic phenomena other than agreement, namely reflexive- and NPI-licensing, and implement this with IO-TSL grammars (Graf and Mayer, 2018): these extend the TSL projection function not only with sensitivity to symbols already output to the tier, but also with sensitivity to local symbols in the input. The syntactic phenomena that Graf and Shafiei address, however, seem to also be within the bounds of the simpler TSL system. As far as

[3] A standard k-TSL grammar is therefore a special case, corresponding to a $(1, k)$-O-TSL grammar.

we are aware then, the omnivorous agreement pattern discussed above presents the most significant challenge to the idea that syntactic constraints are TSL on c-strings, as well as a clear challenge to the more specific idea that this holds for agreement.

7 Conclusion

We are not the first to point out that phonology and syntax– or even vowel harmony and syntactic agreement– seem to make use of similar underlying computational machinery (Nevins, 2010, 2011). Indeed, following the recent surge of work within subregular phonology and syntax, Graf (2022) proposes the *cognitive parallelism hypothesis* that distinct language modules– such as phonology and syntax– have the same formal complexity. What he leaves as an open question, though, is what exactly that level of complexity is.

We've shown here that phonology and syntax each have exceptions, in Uyghur and Mundari respectively, to the claim that each respective module is TSL-2, and that these patterns are exceptional in the same way. In terms of formal complexity, this distinctive exceptionality seems to put each of these patterns just beyond TSL, into O-TSL. But regardless of what exact level of complexity phonology and syntax may share, if there is indeed a similarity between the mechanisms deployed in syntax and phonology, then the formal similarities observed here are the exact kinds we expect to arise.

References

Alëna Aksënova, Thomas Graf, and Sedigheh Moradi. 2016. Morphotactics as tier-based strictly local dependencies. In *Proceedings of the 14th sigmorphon workshop on computational research in phonetics, phonology, and morphology*, pages 121–130.

Susana Béjar. 2003. *Phi-syntax: A theory of agreement*. University of Toronto Toronto, ON.

Jane Chandlee. 2014. *Strictly local phonological processes*. University of Delaware.

Noam Chomsky. 2001. Derivation by phase. In Michael J. Kenstowicz, editor, *Ken Hale: A Life in language*. MIT Press, Cambridge, MA.

Aniello De Santo and Thomas Graf. 2019. Structure sensitive tier projection: Applications and formal properties. In *International conference on formal grammar*, pages 35–50. Springer.

Paul D Fallon. 1993. Liquid dissimilation in georgian. In *Eastern States Conference on Linguistics*, volume 10, pages 105–116.

Thomas Graf. 2022. Subregular linguistics: bridging theoretical linguistics and formal grammar. *Theoretical Linguistics*, 48(3-4):145–184.

Thomas Graf and Connor Mayer. 2018. Sanskrit n-Retroflexion is Input-Output Tier-Based Strictly Local. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 151–160, Brussels, Belgium. 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.

Kenneth Hanson. 2025. Tier-based strict locality and the typology of agreement. *Journal of Language Modelling*, 13(1):43–97.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. In Larry M. Hyman and Frans Plank, editors, *Phonological Typology*, pages 126–195. De Gruyter.

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.

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

Douglas C. Johnson. 1972. *Formal Aspects of Phonological Description*. Monographs in Linguistic Analysis. Mouton, The Hague.

Ronald M Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational linguistics*, 20(3):331–378.

Connor Mayer and Travis Major. 2018. A challenge for tier-based strict locality from uyghur backness harmony. In *International conference on formal grammar*, pages 62–83. Springer.

Gurujegan Murugesan, Imke Driemel, and Andrew Murphy. 2025. Omnivorous person, number and gender in Mundari: A cyclic Agree analysis. *Natural Language & Linguistic Theory*.

Andrew Nevins. 2010. *Locality in vowel harmony*, volume 55. Mit Press.

Andrew Nevins. 2011. Multiple agree with clitics: Person complementarity vs. omnivorous number. *Natural language & linguistic theory*, 29(4):939–971.

Modelling the Diachronic Emergence of Phoneme Frequency Distributions

Fermín Moscoso del Prado Martín
Department of Computer Science and Technology
University of Cambridge, UK
fm611@cst.cam.ac.uk

Suchir Salhan
Department of Computer Science and Technology
University of Cambridge, UK
sas245@cst.cam.ac.uk

Abstract

Phoneme frequency distributions exhibit robust statistical regularities across languages, including exponential-tailed rank-frequency patterns and a negative relationship between phonemic inventory size and the relative entropy of the distribution. The origin of these patterns remains largely unexplained. In this paper, we investigate whether they can arise as consequences of the historical processes that shape phonological systems. We introduce a stochastic model of phonological change and simulate the diachronic evolution of phoneme inventories. A naïve version of the model reproduces the general shape of phoneme rank-frequency distributions but fails to capture other empirical properties. Extending the model with two additional assumptions –an effect related to frequency and a stabilising tendency toward a preferred inventory size– yields simulations that match both the observed distributions and the negative relationship between inventory size and relative entropy. These results suggest that some statistical regularities of phonological systems may arise as a result of diachronic sound change instead of –or in addition to– explicit optimisation or compensatory mechanisms.

1 Introduction

The frequency distributions with which different phonemes occur in a language can provide insights into the nature, representation, and processing of human language. Nevertheless, despite its importance, only a handful of studies have investigated the nature of these distributions; most investigated these distributions from a macroscopic (in the sense of Mandelbrot, 1957) perspective, considering the shape of the rank-frequency plots (Sigurd, 1968; Good, 1969; Martindale et al., 1996; Martindale and Tambovtsev, 2007; Macklin-Cordes and Round, 2020; Moscoso del Prado Martín and Salhan, 2026).

Recently, it has been reported (Moscoso del Prado Martín and Salhan, 2026) that –at both macro- and microscopic levels of description– phoneme frequency distributions exhibit detectable effects consistent with the 'Compensation Hypothesis' (Hockett, 1955; Martinet, 1955). This hypothesis predicts that increased complexity in one domain of language is offset elsewhere in the language. Several studies had previously found evidence for this hypothesis involving balancing aspects of the phonological system with other aspects of language structure (Moran and Blasi, 2014; Pimentel et al., 2020, 2021). It is, however, remarkable that compensation is already detectable by examining the unigram phoneme frequency distributions alone.

An important related question has received little attention: How do phoneme frequency distributions arise in diachronic terms? In particular, one could also ask whether what appear as compensation effects might in fact be unexpected consequences of the historical processes that have shaped the phoneme inventories. This would open the possibility of compensation phenomena being epiphenomenal, rather than the result of any actual optimisation process. Inspired by the predictions of Evolutionary Phonology (Blevins, 2004), Ceolin and Sayeed (2019) and Ceolin (2020) introduce a stochastic split-and-merger model of sound change in which phoneme inventories evolve through repeated splitting and merging events. Their simulations show that statistical patterns commonly associated with phonological markedness can emerge from simple diachronic processes, without assuming markedness as a primitive property of phonological systems. In a similar vein, in this study, we introduce a model of phoneme change based on Hoenigswald's (1965) typology of phonological changes. We use this model to investigate whether two simple diachronic principles –frequency and a stabilising tendency in phoneme inventory size– are sufficient for generating both the observed phoneme rank-

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 138–146
July 3-4, 2026 ©2026 Association for Computational Linguistics

frequency distributions and the negative relationship between phonemic inventory size and relative entropy.

In what follows, we first describe the empirical properties of phoneme frequency distributions that our model aims to explain (Section 2). We then introduce a stochastic model of phonological change (Section 3). We examine three incremental versions of the model of increasing complexity: a naïve baseline (Section 3.3), a version incorporating effects related to frequency (Section 3.4), and a model further introducing a central tendency in phonemic inventory size (Section 3.5). We show that the latter reproduces both the observed rank-frequency patterns of phoneme frequencies and the negative relationship between phonemic inventory size and relative entropy, suggesting that these macroscopic patterns may arise as natural consequences of diachronic phonological dynamics.

2 Properties of Phoneme Distributions

Exponential tails: Contrary to arguments for power-law distributions of phonemes (Sigurd, 1968; Martindale et al., 1996; Martindale and Tambovtsev, 2007; Ceolin and Sayeed, 2019; Ceolin, 2020), phoneme frequency distributions exhibit rank-frequency plots typical of exponential-tailed distributions (Good, 1969; Macklin-Cordes and Round, 2020; Moscoso del Prado Martín and Salhan, 2026): In the double-logarithmic plane, the right tails of these distributions fall abruptly, quickly deviating from the straight lines that are characteristic of power-law-tailed distributions (e.g., word frequency distributions; Condon, 1928). These patterns are illustrated in Figure 1a for the 166 languages of Macklin-Cordes and Round's (2020) dataset of Australian language varieties,[1] and for the 107 languages in the NorthEuraLex database (Dellert et al., 2019).[2]

Correlation between phonemic inventory size and relative entropy: In a recent study, Moscoso del Prado Martín and Salhan (2026) report that –across the world's languages– there is a negative correlation between a language's *Phonemic Inventory Size* (PIS), and the relative entropy (Shannon, 1948) of its phoneme distribution. The entropy of the phoneme distribution is an indicator of the per-phoneme informational content of a language (Cherry et al., 1953). Generally speaking, increasing the PIS increases the potential value of the distribution's entropy. However, the reduction in relative entropy attenuates the effect of increasing the PIS on the inventory's entropy, as would be predicted by the Compensation Hypothesis (Hockett, 1955; Martinet, 1955). This is illustrated in Figure 1b. The figure plots the negative correlation between PIS and relative entropy across the languages in the Australian languages and NorthEuraLex datasets mentioned above. Note that this relation is rather robust, it is present on the Australian language dataset –which was one of the datasets used by Moscoso del Prado Martín and Salhan–, however, the relation also extends to the NorthEuraLex languages –not used in the previous study. In fact, as shown in the figure, the precise prediction made by Moscoso del Prado Martín and Salhan (dashed line in the figure) extrapolates into the NorthEuraLex observations remarkably well.

3 Modelling Phonological Change

3.1 Phonological changes

Phonological changes (also known as phonemic changes) are diachronic changes within a language's system of contrastive sounds. Such changes go beyond mere shifts in pronunciation, which are referred to as *phonetic changes*. They affect the oppositions among phonemes, and may result in changes to a language's phoneme inventory: Contrasts may arise, disappear, or be reorganised, resulting in a restructuring of the language's sound system.

Phonological changes can be of different types. Using Hoenigswald's (1965) systematisation, we can distinguish between three main types: *Primary splits* (also known as conditioned mergers) occur when some instances of a phoneme A become an already existing phoneme B in particular contexts. In these cases, the number of phonemes remains the same, but their distribution changes. *Secondary splits* (also known as phonemic splits) occur when some instances of A become a new phoneme B. Changes of this type result in new contrast being created. In turn this increases the size of phonemic inventories. Finally, *unconditioned mergers* (also known simply as mergers) occur when all instances of phonemes A and B collapse into a single phoneme A. In these cases, a contrast is eliminated and the number of phonemes decreases. Note that,

[1] Downloaded from https://zenodo.org/records/4104116 on May 1, 2025.

[2] v0.9, downloaded from the LexiBank Project at https://github.com/lexibank/northeuralex on March 3, 2026.

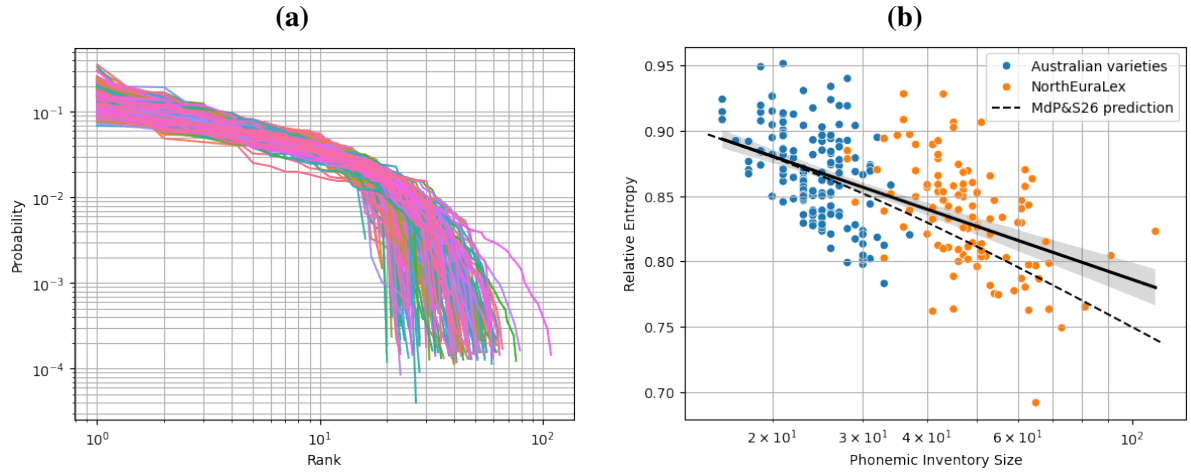


Figure 1: **(a)** Rank-frequency plots (note the log-log scale) for the phoneme frequency distributions across the languages in Macklin-Cordes and Round's (2020) dataset of Australian language varieties, and those in NorthEuraLex. Each line plots one language. **(b)** Relationship between PIS and relative entropy for the languages in Macklin-Cordes and Round's (2020) dataset of Australian language varieties, and those in NorthEuraLex. Each point plots one language. The solid line plot a log-linear regression, and the shading plots its 95% C.I.. The dashed line plots the relation as predicted by Moscoso del Prado Martín and Salhan (2026).

within this classification, there is no specific type for phoneme losses; these are just considered unconditional mergers with a zero phoneme.

3.2 The general model

Consider language at a particular time point τ in its history. It uses V_τ distinct contrastive phonemes $\pi_1, \pi_2, \ldots, \pi_{V_\tau}$. The probabilities with which each phoneme occurs in the language at time τ are given by the vector $\mathbf{p}_\tau = \{\mathrm{p}_\tau(\pi_1), \mathrm{p}_\tau(\pi_2), \ldots, \mathrm{p}_\tau(\pi_{V_\tau})\}$, where $\sum_{i=1}^{V_\tau} \mathrm{p}_\tau(\pi_i) = 1$. Time is treated as a discrete sequence of phonological changes c_τ. Each change is of one of three types: primary split (p), secondary split (s), or merger (m). Formally, c_τ is sampled at each time step from the alphabet $\Sigma = \{p, s, m\}$. Time intervals are defined so that each contains exactly one change event.

We construct stochastic models in which, at every time point τ, the change type c_τ is sampled randomly from Σ, with probabilities $P_\tau(p)$, $P_\tau(s)$, and $P_\tau(m)$, so that $P_\tau(p) + P_\tau(s) + P_\tau(m) = 1$. Independently, a phoneme $\pi_i \in \{\pi_1, \ldots, \pi_{V_\tau}\}$ is sampled (according to some predefined scheme, details below) as the target of change. In the case of primary splits and mergers, a second phoneme π_j is also sampled, representing the phoneme toward which π_i shifts (primary split) or with which it merges (merger). In addition, a proportion parameter $\alpha_\tau \in (0, 1]$ is sampled to determine how much probability mass is transferred between phonemes. We assume α_τ is uniformly distributed, reflecting the fact that phonological changes can affect contrasts to varying degrees.

The three types of changes are defined as transformations on the probability vector:

Primary split: a proportion α_τ of $\mathrm{p}(\pi_i)$ is reassigned to an existing phoneme π_j:

$$\begin{aligned} \mathrm{p}_{\tau+1}(\pi_i) &= (1 - \alpha_\tau)\mathrm{p}_\tau(\pi_i), \\ \mathrm{p}_{\tau+1}(\pi_j) &= \mathrm{p}_\tau(\pi_j) + \alpha_\tau\, \mathrm{p}_\tau(\pi_i). \end{aligned} \tag{1}$$

For $0 < \alpha_\tau < 1$, the inventory size remains $V_{\tau+1} = V_\tau$; if $\alpha_\tau = 1$, π_i disappears and $V_{\tau+1} = V_\tau - 1$.

Secondary split: a new phoneme $\pi_{V_\tau+1}$ is created by splitting π_i:

$$\begin{aligned} \mathrm{p}_{\tau+1}(\pi_i) &= (1 - \alpha_\tau)\mathrm{p}_\tau(\pi_i), \\ \mathrm{p}_{\tau+1}(\pi_{V_\tau+1}) &= \alpha_\tau\, \mathrm{p}_\tau(\pi_i). \end{aligned} \tag{2}$$

For $0 < \alpha_\tau < 1$, $V_{\tau+1} = V_\tau + 1$; if $\alpha_\tau = 1$, π_i disappears and $V_{\tau+1} = V_\tau$

Unconditioned merger: two phonemes π_i and π_j collapse completely:

$$\mathrm{p}_{\tau+1}(\pi_j) = \mathrm{p}_\tau(\pi_j) + \mathrm{p}_\tau(\pi_i), \tag{3}$$

and π_i is removed, therefore $V_{\tau+1} = V_\tau - 1$.

Over time, this process generates stochastic trajectories both in inventory size V_τ and in the internal distributional structure of the phonological

system (the probability vector). As we show below, the method by which the change probabilities $P_\tau(p)$, $P_\tau(s)$, and $P_\tau(m)$, the choice of phonemes, and the α_τ are chosen at each time point can substantially affect the behaviour of these models. In summary, this model treats phonological change as a stochastic process redistributing probability mass across phonemes while allowing contrasts to be created or eliminated.

3.3 Simulation 1: a naïve model

As a baseline, we simulated the basic model in Section 3.2, using its simplest instantiation. We considered constant equal probabilities $P_\tau(p) = P_\tau(s) = P_\tau(m) = 1/3$ for all timepoints τ. At each time step, the phonemes involved in the change (π_i and π_j in the notation above) were randomly sampled from the current phoneme inventory with uniform probabilities (i.e., any phoneme was equally likely to be chosen), and the proportion of phonemes involved in the change (α_τ), when necessary, was sampled from a uniform [0,1] distribution. We simulated the evolution of 400 distinct languages, each for 1,000 time steps. Initially, all languages were set to uniform distributions of 34 phonemes (this number was taken as the approximate mean PIS for the world's languages according to PHOIBLE 2.0; Moran and McCloy, 2019).

As shown in Figure 2a, even this naïve simulation already generates rank-frequency patterns resembling those observed in empirical phoneme distributions (Figure 1a). This is so even when all distributions started out as uniforms. However, examining the simulated distributions in more detail reveals crucial differences with real phoneme frequency distributions. Figure 2b plots the correlation between (log) PIS and the relative entropy of the distribution, which is significant (Pearson's $r = .47$, $p < .01$). However, this goes in exactly the opposite direction than it did in real phoneme distributions: Whereas simulated data show a positive correlation, in actual distributions it is negative. This difference indicates that, in order to capture the properties of phoneme frequencies, our diachronic models need to consider further details.

3.4 Simulation 2: considering frequency

Simulation 1 was too naïve in assuming that all phonemes are equally likely to take part in phonological change. This is in fact known not to be the case. The *Functional Load Hypothesis* –tracing back to Gilliéron (1918)– proposes that phonological contrasts that distinguish many words (i.e., have high functional load) are less likely to be lost through sound change, whereas contrasts that contribute little to lexical differentiation are more prone to merger (for details see, Hockett, 1955, 1967). This hypothesis has received empirical support (Wedel et al., 2013a,b). Our models are too coarse-grained to explicitly consider a direct measure of functional load (see Surendran and Niyogi, 2003). We can however, indirectly take it to account to some degree: Wedel et al. (2013b) showed that –in general– a phoneme's functional load is positively correlated with its frequency of occurrence.

In order to take phoneme frequency into consideration, we ran new simulations. As before, we considered constant equal probabilities $P_\tau(p) = P_\tau(s) = P_\tau(m) = 1/3$ for all timepoints τ, and the proportion of phonemes involved in the change (α_τ), when necessary, was sampled from a uniform [0,1] distribution. Again, we simulated the evolution of 400 distinct languages, each for 1,000 time steps. Initially, all languages were set to uniform distributions of 34 phonemes.

This time, however, at each time step, the phonemes involved in the change were randomly sampled from the current phoneme inventory with different probabilities. The phonemes that would be split or merged into another one (i.e., the phonemes whose probabilities are decreased in Equations 1, 2, and 3) were sampled with probabilities directly proportional to their surprisals (i.e., less frequent phonemes were sampled more often). On the other hand, the phonemes whose probabilities would increase (in Equations 1 and 3) were sampled uniformly (i.e., all phonemes are equally likely to be chosen). In this way, we have increased the probability that the less frequent phonemes are lost in historical change.

The resulting distributions retain the characteristic shape of phoneme rank-frequency curves observed in real languages, albeit with a substantially increased variance; the introduction of the frequency bias causes a rich-get-richer effect, resulting in many extremely skewed distributions with close to zero relative entropies (see Figure 3a). The modification did not fix the incorrect sign of the PIS–relative entropy correlation (Pearson's $r = .68$, $p < .01$; see Figure 3b).

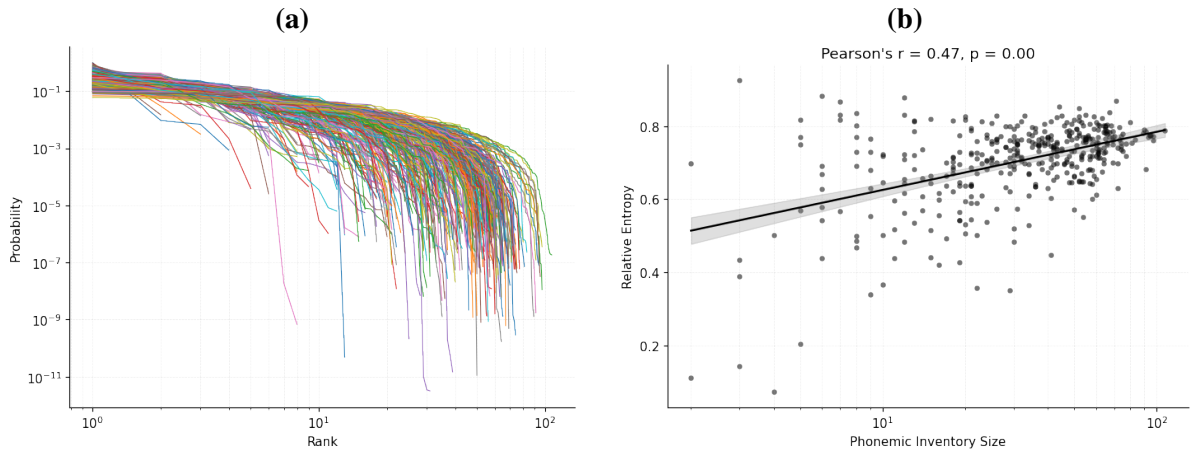


Figure 2: **(a)** Rank-frequency plots (note the log-log scale) for the final phoneme frequency distributions in Simulation 1. Each line plots one language. **(b)** Relationship between PIS and relative entropy for the final distributions in Simulation 1. Each point plots one simulated language. The solid line plots a log-linear regression, and the shading plots its 95% C.I..

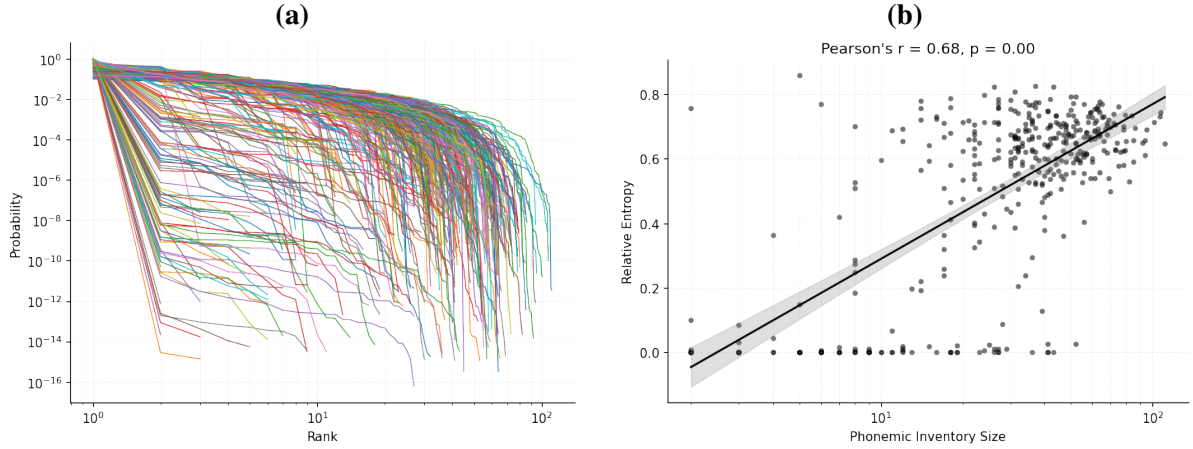


Figure 3: **(a)** Rank-frequency plots (note the log-log scale) for the final phoneme frequency distributions in Simulation 2. Each line plots one language. **(b)** Relationship between PIS and relative entropy for the final distributions in Simulation 2. Each point plots one simulated language. The solid line plots a log-linear regression, and the shading plots its 95% C.I..

3.5 Simulation 3: adding a central tendency

Simulation 2 still presents two problems with respect to actual phoneme frequency distributions. First, it does not show the correct relationship between PIS and relative entropy. By itself, this might not be so much of a problem. It could very well be that this correlation truly arises as a consequence of actual compensation. Indeed, Moscoso del Prado Martín and Salhan (2026) find that such correlation could be explained microscopically by compensation at the level of phonotactics, by which languages with larger inventories tend to have phonemes that appear in more predictable contexts. As our models do not include any microscopic details on phonotactics, it could be the case that we are not able to model the emergence of compensation.

Second, and more critically, in Simulations 1 and 2, PIS values evolve as random walks on the positive integers. At each step, the step size probabilities depend on the probabilities of the three change types. In the idealised case in which $0 < \alpha_\tau < 1$ almost surely,

$$\begin{aligned} V_{\tau+1} - V_\tau &= \Delta V_\tau, \\ P(\Delta V_\tau = 0) &= P_\tau(p), \\ P(\Delta V_\tau = +1) &= P_\tau(s), \\ P(\Delta V_\tau = -1) &= P_\tau(m). \end{aligned} \tag{4}$$

If α_τ can take the boundary value 1, primary splits and secondary splits may also reduce or preserve

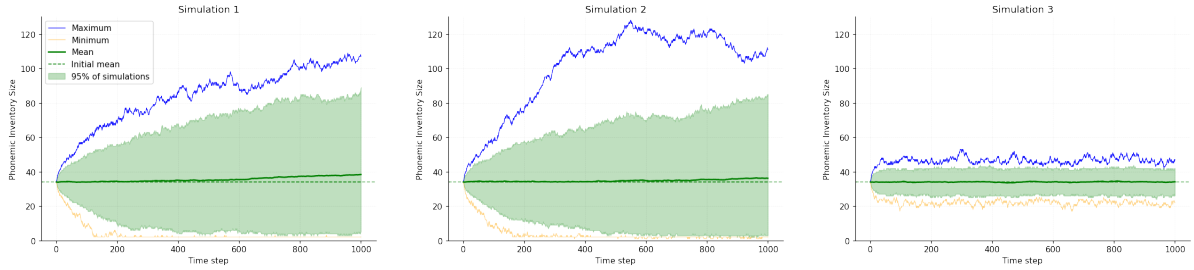


Figure 4: Convergence of the PIS in the simulations. The blue and orange lines respectively plot the maximum and minimum values at each time point across the 400 simulated languages. The solid green lines plot the mean PIS at each time point, the dashed green lines are the predicted mean μ, and the shaded areas contain 95% of the runs in each of the simulations.

V_τ through elimination of the source phoneme. Moreover, mergers are only defined when V_τ exceeds a minimal value, which induces a reflecting constraint at small inventories. Whether this random walk has a stationary mean or it has a drift depends on the specific values of the different parameters. However, as long as the probabilities $P_\tau(p)$, $P_\tau(s)$, and $P_\tau(m)$ remain constant in time, the *variance* of V_τ is bound to increase with time. As a consequence, in this situation, as time goes on, there is no limit to the size that V_τ can take, and the minimum goes all the way down to two phonemes.[3] The resulting random walks for Simulations 1 and 2 are summarised in Figure 4. Notice that in both simulations, even after 1,000 time steps the ranges of PIS values keep increasing without any upper bounds.

This property is not desirable for a model of phonological change. First, there is no evidence that the range of values in the PIS is increasing with time in the world's languages. Previous work has noted that the number of phonemic contrasts across languages is concentrated within a relatively narrow range. Extremely small or large inventories are rather rare (Maddieson, 1984; Anderson et al., 2023). For instance, according to Phoible v2.0 (Moran and McCloy, 2019), the minimum value of the PIS is that of some variants of Rotokas, which may have as few as eleven contrasts (if vowel length is not considered), and the maximum is capped at the 160 contrasts attributed to East Taa. Were the range of PIS left to increase freely in an unconstrained random walk, we would expect more extreme values to be observed among thousands of languages. Also as a result, as can be appreciated comparing Figure 1a with Figures 2a and 3a, the variability in the lower rank phonemes is substantially larger in the simulations than in the real language data.

The only way of avoiding such growing variance of the V_τ is to make the P_τ values dependent on the value of V_τ itself. We assume there is some optimal value of V_τ, which we denote by μ. The further V_τ is from μ, the less likely it should be to get even further. From Equation 4 we see that $P_\tau(s)$ should decrease in value when $V_\tau > \mu$, and in turn, whenever $V_\tau < \mu$, it is $P_\tau(m)$ that should be lowered. We implemented this adaptive strategy using exponential functions, implementing a smooth bias toward μ while preserving positive probabilities,

$$\begin{aligned} P_\tau(p) &= \frac{1}{k(\tau)} \\ P_\tau(s) &= \frac{1}{k(\tau)}\, \mathrm{e}^{(\mu - V_\tau)/\mu} \\ P_\tau(m) &= \frac{1}{k(\tau)}\, \mathrm{e}^{(V_\tau - \mu)/\mu} \\ k(\tau) &= 1 + \mathrm{e}^{(\mu - V_\tau)/\mu} + \mathrm{e}^{(V_\tau - \mu)/\mu} \end{aligned} \tag{5}$$

We modified Simulation 2 to include the adaptive scheme above, with $\mu = 34$ (the mean PIS across the world's languages). As before, we simulated 400 languages for 1,000 time steps. Figure 4 shows that the simulations have now converged in terms of maximum, minimum and range of values of the PIS.[4]

Figure 5a plots the rank-frequency distributions of the simulated inventories. Notice that the variability on the lower ranks has considerably decreased in relation to Simulations 1 and 2 (compare with Figures 2a and 3a), the distributions are more

[3] This was set as a hard limit in the simulations, as it wouldn't make any sense to have languages with a single contrastive phoneme.

[4] Note that, as seen is the results of Simulations 1 and 2, convergence in terms of the shape of the rank-frequency plot is achieved in all cases, whether the PIS has converged or not.

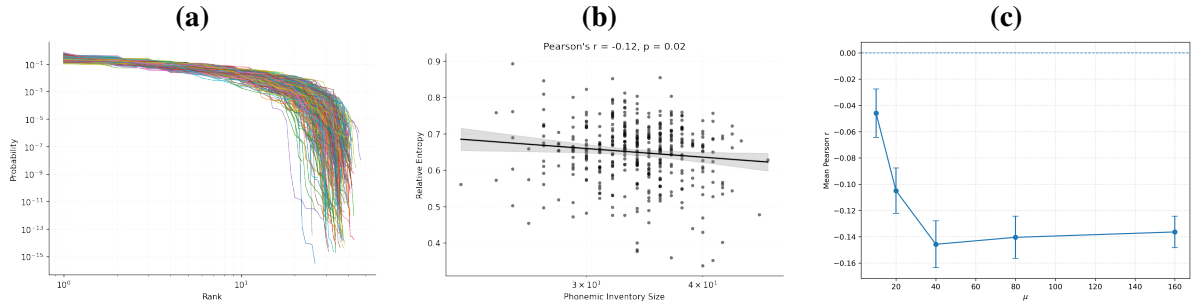


Figure 5: **(a)** Rank-frequency plots (note the log-log scale) for the final phoneme frequency distributions in Simulation 3. Each line plots one language. **(b)** Relationship between PIS and relative entropy for the final distributions in Simulation 3. Each point plots one simulated language. The solid line plots a log-linear regression, and the shading plots its 95% C.I.. **(c)** Sensitivity analysis as for parameter μ: mean and standard error of the Pearson correlation coefficient between PIS and relative entropy for different values of μ across ten simulations for each of the μ values.

similar to those observed in real languages (compare with Figure 1a). Most crucially, as Figure 5b plots, a significant negative correlation between PIS and relative entropy has now emerged (Pearson's $r = -.12$, $p = .02$). In short, both of the problems in the previous simulations have been solved by introducing the central tendency. Admittedly, this is a very slight correlation, substantially weaker than that obtained for real languages. Nevertheless, it is quite robust: Repeating the simulation 100 times resulted in a negative r value every single time ($\bar{r} = -.15 \pm .005$), and these correlations reached significance in 78 out of the 100 runs. In addition, as shown by the sensitivity analysis summarised in Figure 5c, this negative correlation seems to arise irrespective of the value of the central tendency parameter μ.

One could fit the different simulation parameters to make the resulting values match human languages more closely, but this is not necessary for the general patterns to arise. Remarkably, the negative relationship between PIS and relative entropy can arise without any actual compensation mechanisms being involved. In our simulations, it is just an unexpected side effect of the interaction between phonological changes, and a central tendency towards a specific number of contrasts. In this sense, what appears as compensation could in fact be epiphenomenal.

4 Discussion

The simulations presented here show that several macroscopic properties of phoneme frequency distributions can arise from relatively simple diachronic mechanisms. In particular, stochastic phonological change acting on phoneme inventories naturally generates rank-frequency distributions with exponential tails similar to those observed in real languages. Such distributions arise naturally from the historical processes that continually reshape phonological systems. Our results should be interpreted as providing a generative explanation for the statistical patterns. The goal of the models is not to capture all mechanisms involved in phonological change. Rather, we test whether simple diachronic processes are sufficient to generate the observed macroscopic regularities of phoneme frequency distributions.

Naïve versions of the model fail to capture important empirical patterns. In particular, they predict a positive relationship between phonemic inventory size (PIS) and relative entropy –contrary to the findings of Moscoso del Prado Martín and Salhan (2026)– and they produce unbounded variation in inventory sizes over time. Introducing a central tendency in the evolution of PIS substantially improves the models. This constrains the random walk behaviour of inventory size, producing stationary dynamics that are more consistent with the observed range of phoneme inventories across languages.

Not including a central attractor –as in Simulations 1 and 2, or in the models of Ceolin and Sayeed (2019) and Ceolin (2020)– results in phonemic inventory sizes that are too broadly distributed. Instead, our assumption of a central tendency in phonemic inventory size is consistent with previous research on phonological systems. Typological surveys have long noted that the number of phonemic contrasts across languages is concentrated within

a relatively narrow range, with extremely small or extremely large inventories being rare (Maddieson, 1984; Anderson et al., 2023). At a theoretical level, models such as Adaptive Dispersion Theory propose that phonological systems evolve under competing pressures for perceptual distinctiveness and articulatory economy, which tend to produce stable configurations of contrasts (Liljencrants and Lindblom, 1972; Lindblom, 1986; Lindblom and Maddieson, 1988). From this perspective, a stabilising tendency in the evolution of phonemic inventories can be interpreted as a coarse-grained reflection of underlying pressures shaping phonological systems. Our model does not attempt to represent these mechanisms explicitly, but instead captures their aggregate effect through a simple probabilistic bias toward a preferred inventory size.

5 Conclusion

Taken together, these results suggest that some statistical properties of phonological systems may emerge from the interaction of diachronic pressures –in line with the results of Ceolin and Sayeed (2019) and Ceolin (2020). In particular, the negative relationship between PIS and relative entropy was previously interpreted as evidence for compensatory organisation within the phonological system (Moscoso del Prado Martín and Salhan, 2026). However, it could also arise –perhaps in part– as a by-product of stochastic sound change operating under a stabilising tendency toward a preferred inventory size. In this possibility, rather than balancing complexity across organisational tiers, compensation might arise in diachronic time from balancing pressures between perception and articulation. This does not rule out the possibility that genuine compensatory mechanisms between tiers could exist. However, it highlights the importance of considering whether apparent optimisation effects may instead reflect epiphenomenal consequences of the dynamics of sound change, much in the spirit of Evolutionary Phonology (Blevins, 2004, 2019).

References

Cormac Anderson, Tiago Tresoldi, Simon J. Greenhill, Robert Forkel, Russell D. Gray, and Johann-Mattis List. 2023. Variation in phoneme inventories: Quantifying the problem and improving comparability. *Journal of Language Evolution*, 8(2):149–168.

Juliette Blevins. 2004. *Evolutionary phonology: The emergence of sound patterns*. Cambridge University Press, Cambridge, UK.

Juliette Blevins. 2019. Evolutionary phonology as human behavior. In Nancy Stern, Ricardo Otheguy, Wallis Reid, and Jaseleen Sackler, editors, *Columbia School Linguistics in the 21st Century*, pages 281–300. John Benjamins Publishing Company.

Andrea Ceolin. 2020. *Functionalism, Lexical Contrast and Sound Change*. Ph.D. thesis, University of Pennsylvania.

Andrea Ceolin and Ollie Sayeed. 2019. Modeling markedness with a split-and-merger model of sound change. In Nina Tahmasebi, Lars Borin, Adam Jatowt, and Yang Xu, editors, *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change*, pages 67–70. Association for Computational Linguistics, Florence, Italy.

E. Colin Cherry, Morris Halle, and Roman Jakobson. 1953. Toward the logical description of languages in their phonemic aspect. *Language*, 29(1):34–46.

Edward Uhler Condon. 1928. Statistics of vocabulary. *Science*, 67(1733):300.

Johannes Dellert, Thora Daneyko, Alla Münch, Alina Ladygina, Armin Buch, Natalie Clarius, Ilja Grigorjew, Mohamed Balabel, Hizniye Isabella Boga, Zalina Baysarova, Roland Mühlenbernd, Johannes Wahle, and Gerhard Jäger. 2019. NorthEuraLex: a wide-coverage lexical database of Northern Eurasia. *Language Resources and Evaluation*, 54(2):273–301.

Jules Gilliéron. 1918. *Généalogie des Mots qui Désignent l'Abeille d'après l'Atlas Linguistique de la France [The Genealogy of the Words for 'Bee' According to the Linguistic Atlas of France]*, volume 225 of *Bibliothèque de l'École des Hautes Études: Sciences Historiques et Philologiques*. Librairie Ancienne Honoré Champion, Paris.

Irving John Good. 1969. Statistics of language. In Andrew R. Meetham and Richard A. Hudson, editors, *Encyclopaedia of Linguistics, Information and Control*, pages 567–581. Pergamon Press, Oxford, England.

Charles F. Hockett. 1955. *A Manual of Phonology*. Waverly Press, Baltimore, MD.

Charles F. Hockett. 1967. The quantification of functional load. *Word*, 23(1–3):320–339.

Henry M. Hoenigswald. 1965. *Language Change and Linguistic Reconstruction*. University of Chicago Press, Chicago, IL.

Johan Liljencrants and Björn Lindblom. 1972. Numerical simulation of vowel quality systems: The role of perceptual contrast. *Language*, 48(4):839–862.

Björn Lindblom. 1986. Phonetic universals in vowel systems. In John J. Ohala and Jeri J. Jaeger, editors, *Experimental Phonology*, pages 13–44. Academic Press, Orlando, FL.

Björn Lindblom and Ian Maddieson. 1988. Phonetic universals in consonant systems. In Charles N. Li and Larry M. Hyman, editors, *Language, Speech and Mind*, pages 62–78. Routledge, London.

Jayden L. Macklin-Cordes and Erich R. Round. 2020. Re-evaluating phoneme frequencies. *Frontiers in Psychology*, 11:570895.

Ian Maddieson. 1984. *Patterns of Sounds*. Cambridge University Press, Cambridge, England.

Benoît B. Mandelbrot. 1957. Linguistique statistique macroscopique [Macroscopic statistical linguistics]. In Léo Apostel, Benoît B. Mandelbrot, and Albert Morf, editors, *Logique, Langage, et Théorie de l'Information*. Presses Universitaires de France, Paris, France.

Colin Martindale, Sabir M. Gusein-Zade, Dean McKenzie, and Mark Yu. Borodovsky. 1996. Comparison of equations describing the ranked frequency distributions of graphemes and phonemes. *Journal of Quantitative Linguistics*, 3:106–112.

Colin Martindale and Yuri Tambovtsev. 2007. Phoneme frequencies follow a Yule distribution. *SKASE Journal of Theoretical Linguistics*, 4:1–11.

André Martinet. 1955. *Économie des Changements Phonétiques [Economy of Phonetic Changes]*. A. Francke, Bern, Switzerland.

Steven Moran and Damián Blasi. 2014. Cross-linguistic comparison of complexity measures in phonological systems. In Frederick J. Newmeyer and Laurel B. Preston, editors, *Measuring Grammatical Complexity*, pages 217–240. Oxford University Press, Oxford, England.

Steven Moran and Daniel McCloy, editors. 2019. *PHOIBLE 2.0*. Max Planck Institute for the Science of Human History, Jena, Germany. Accessed on 04/02/2026.

Fermín Moscoso del Prado Martín and Suchir Salhan. 2026. The distribution of phoneme frequencies across the world's languages: Macroscopic and microscopic information-theoretic models. *Preprint*, arXiv:2603.02860.

Tiago Pimentel, Clara Meister, Elizabeth Salesky, Simone Teufel, Damián Blasi, and Ryan Cotterell. 2021. A surprisal–duration trade-off across and within the world's languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 949–962, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tiago Pimentel, Brian Roark, and Ryan Cotterell. 2020. Phonotactic complexity and its trade-offs. *Transactions of the Association for Computational Linguistics*, 8:1–18.

Claude E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Bengt Sigurd. 1968. Rank-frequency distributions for phonemes. *Phonetica*, 18:1–15.

Dinoj Surendran and Partha Niyogi. 2003. Measuring the functional load of phonological contrasts. Technical Report TR-2003-12, University of Chicago, Department of Computer Science.

Andrew Wedel, Scott A. Jackson, and Abby Kaplan. 2013a. Functional load and the lexicon: Evidence that syntactic category and frequency relationships in minimal lemma pairs predict the loss of phoneme contrasts in language change. *Language and Speech*, 56(3):395–417.

Andrew Wedel, Abby Kaplan, and Scott A. Jackson. 2013b. High functional load inhibits phonological contrast loss: A corpus study. *Cognition*, 128(2):179–186.

Morpheme structure phonotactics: a categorical model for morpho-phonological productivity in Russian vowel-zero alternations

Daniar Kasenov
New York University
dk4981@nyu.edu

Abstract

Nonce word studies motivate a notion of gradient *similarity* between nonce words and real words. In morpho-phonological research, similarity is often taken as to be a relationship between a nonce word and the list of morphemes / words that undergo a given morpho-phonological alternation (Albright and Hayes 2003; Becker et al. 2011 i.a.). This paper challenges this view on the basis of nonce word data on Russian vowel–zero alternations (Gouskova and Becker 2013; Becker and Gouskova 2016). I propose a model where morpho-phonological similarity is a relationship between the available underlying representations and the underlying representation the nonce item must have in order to undergo the alternation. The implementation of the proposed model matches—and in some comparisons exceeds—the performance of Becker and Gouskova's (2016) MaxEnt-model. This study thus presents a linking hypothesis between nonce word studies and approaches that mark segments themselves as undergoing certain restricted alternations.

1 Introduction

Some phonological alternations only apply to certain morphemes (they are *morpho*-phonological: see Zimmermann 2026 for an overview). Such alternations show gradient productivity: their rate of application to nonce words may lie anywhere between 0 and 1 depending on the properties of the nonce word (Albright and Hayes 2003; Hayes et al. 2009; Becker et al. 2011; Becker and Gouskova 2016 i.a.). Consider Russian vowel-zero alternations like *mox* ∼ *mx-a* 'moss', compared to *nos* ∼ *nos-a* 'nose' where the alternation does not apply (Scheer 2011; Gouskova 2012 i.a.). A Russian speaker is more likely to accept a nonce word alternation like *posot*∼*post-a* than an alternation like *posort*∼*posrt-a*, due to syllable structure related restrictions (Gouskova and Becker 2013).

In line with most work in phonotactics, these findings have been interpreted as evidence that linguistic knowledge provides a inherently gradient notion of *similarity* between nonce words and real words, often modelled using a weighted constraints grammar (Hayes and Wilson 2008; Wilson and Gallagher 2018; Mayer and Nelson 2020; Dai et al. 2023; Mayer 2025). For example, the proposal of Becker and Gouskova (2016) relies on Hayes and Wilson's MaxEnt-based phonotactic model.

For morpho-phonological purposes, similarity is often defined as how similar the nonce word is to real words that undergo a morpho-phonological process (so-called sublexicons: Becker et al. 2011; Gouskova and Becker 2013). On this basis, nonce word studies are taken to support theories that mark whole morphemes, rather than particular segments as exceptional targets or triggers of morpho-phonological processes (Gouskova 2012).

This paper challenges these views on the basis of Russian vowel-zero alternations. I examine whether the experimental results of Becker and Gouskova (2016) require a weighted constraints-based phonotactic model or morpheme marking of exceptionality. First, I implement Becker and Gouskova's proposal using a phonotactic model where constraints are equally weighted—to probe whether constraint weights do any real work in predicting the results (see Gorman 2013; Kostyszyn and Heinz 2021; Durvasula 2026 for similar skepticism vis-à-vis phonotactic studies).

Second, I entertain a model for morpho-phonological productivity that relies on evaluating possible underlying representations of a nonce word relative to the underlying representations that already exist in the lexicon. I dub the model *morpheme structure phonotactics*: the speakers are able to evaluate possible URs for a nonce word based on their phonotactic properties relative to real URs, echoing morpheme structure constraints used in phonological theory (Halle 1959; Booij 2011).

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 147–159
July 3-4, 2026 ©2026 Association for Computational Linguistics

As noted by an anonymous reviewer, the proposal is similar to existing work on Dutch voicing alternations: Ernestus and Baayen (2003) find that surrounding segments are a major predictor of whether a nonce word-final obstruent will alternate. In their own words, "speakers are more likely to choose that phoneme as the underlying representation when there are more phonologically/phonetically similar morphemes in the lexicon sharing that phoneme" (ibid.: 29)—this paper extends such attitude to cases of abstract phonemes (since segment-marking of exceptionality often relies on contrasts that are neutralized on the surface).

All three models end up with a similar fit to the nonce word judgement data. On the other hand, theoretical morpho-phonological work has established a number of cases where segments themselves must be the targets of morpho-phonological processes (see Martínez 2008 on Modern Hebrew spirantization; Rubach 2013, 2016 on Polish vowel-zero alternations; Round 2017 on Yidiny final vowel deletion *i.a.*). Thus, while other sources of evidence favor segment-based view of morpho-phonology, nonce word studies do not provide evidence against it—the resulting picture is then favorable for the segment-based view.

The paper is structured as follows. Section 2 introduces two competing models for morpho-phonological productivity that rely on phonotactics. Section 3 introduces the issues surrounding gradience in phonotactic modelling. Section 4 presents the comparison of the models on the dataset of Becker and Gouskova (2016). Section 5 concludes.

2 Morpho-phonological models

Morpho-phonological alternations such as the Russian vowel–zero pattern raise the issue of encoding: since the alternations are exceptional, the lexicon must mark whether a morpheme undergoes the alternation or not. The opposing positions are morpheme- and segment-marking of exceptionality (see Gouskova 2012 for an overview of the debate). Coming back to the example from earlier, the question is whether a feature that conditions vowel deletion, call it F, is a property of the whole morpheme: *mox*∼*mx-a* 'moss' is +F whereas *nos*∼*nos-a* isn't, or a property of the vowel itself: the vowel *o* of *mox*∼*mx-a* 'moss' is underlyingly +F whereas the vowel *o* of *nos*∼*nos-a* isn't. Note that the latter view is often committed to positing "abstract" contrasts that do not surface phonetically.

Nonce word productivity results has been interpreted as evidence in favor of morpheme-marking of exceptionality (Becker et al. 2011; Becker and Gouskova 2016 *i.a.*). The argument goes: phonotactic tendencies that only hold of items that undergo an exceptional alternation predict whether a nonce word can undergo that alternation. Therefore, the full segmental composition of the morphemes that undergo an alternation must be available for the speaker to analogize over in a nonce word task.

In this section, I outline two models for morpho-phonological productivity. The first model is the established sublexical model (Becker et al. 2011; Allen and Becker 2015; Becker and Gouskova 2016), and relies on morpheme-marking of exceptionality. The other model is the *morpheme structure phonotactics* model, loosely inspired by Ernestus and Baayen (2003): phonotactic tendencies in morpho-phonological productivity are due to distribution of underlying segments in the lexicon. The question, then, becomes a question of locality: whether the immediate surrounding environment of the segment targeted by an exceptional alternation suffices to capture the phonotactic tendencies. I elaborate on the problem in Section 2.3.

2.1 A morpheme exceptionality model

Many theorists reify the observation about morphemes (not) undergoing an alternation: the phonological alternation is posited to apply only if the vowel is a part of a morpheme marked with a diacritic feature z. This idea makes it part of linguistic knowledge whether any given morpheme undergoes the alternation (Pater 2008; Gouskova 2012). E.g., the morpheme *konets* 'end.NOM.SG' that surfaces without the second vowel in *konts-a* 'end-GEN.SG' will bear z while the non-alternating *master* 'master.NOM.SG' ∼ *master-a* 'master-GEN.SG' will not. The alternation itself is driven by a markedness constraint like *MID$_z$ which penalizes mid vowels /e,o/ in realizations of morphemes marked by z, favoring vowel deletion (Gouskova 2012).

Such analysis makes available a list of morphemes that are marked with z, a sublexicon. Gouskova and Becker (2013) show that the sublexicon with alternating vowels has phonotactic tendencies that do not hold of the lexicon as a whole. For example, words that end with a CC sequence like *most* 'bridge' are 17% of masculine nouns in Russian, **no** masculine nouns with an alternating vowel end with a CC sequence. Nonce word tasks show that speakers are aware of this fact (ibid.).

	*CC]	*MID$_z$	MAX
a. /nos-a/			
☞ [nos-a]			
[ns-a]			*!
b. /mox$_z$-a/			
☞ [mx-a]			*
[mox-a]		*!	
c. /mox$_z$/			
☞ [mox]		*	
[mx]	*!		*

Figure 1: The basic logic of Gouskova's analysis: MID$_z$ over MAX forces deletion which is blocked by high ranking *CC] constraint.

The sublexical morpho-phonological model (Becker et al. 2011; Allen and Becker 2015; Becker and Gouskova 2016) makes use of the idea that the speaker's knowledge of the alternation triggered by the feature z involves phonotactic generalizations over the sublexicon of items that bear the feature z. Nonce word task is then a matter of assigning the feature z to a nonce word—thus, evaluating whether the it conforms to existing the generalizations about z-bearing items, i.e. how similar the nonce word is to the list of z-bearing words.

$$p(nom \sim gen_\emptyset) \propto s(nom \mid L_z) \times s(gen_\emptyset \mid L)$$

How to read the proportion: $p(nom \sim gen_\emptyset)$, the probability of accepting a form of the nonce item with vowel deletion, is proportional to the product of $s(gen_\emptyset \mid L)$, the measure of phonotactic similarity of the form with vowel deletion $gen_\emptyset$ to the whole lexicon L, and $s(nom \mid L_z)$, the measure of phonotactic similarity of the base form nom to the sublexicon of forms bearing the feature z, L_z—in other words, whether the derived form conforms to the phonotactics of the whole language and the base conforms to the phonotactics of the sublexicon.

To generalize, $p(B \sim D)$, the probability of the alternation $B \sim D$ between the base B and the derived form D, is proportional to: $s(D \mid L)$, the measure of phonotactic similarity of D to the lexicon L, $s(B \mid L_z)$, the measure of phonotactic similarity of B to a sublexicon L_z, and $p(B \mapsto_z D)$ the probability of B corresponding to D given the diacritic z.

$$p(B \sim D) \propto s(B \mid L_z) \times s(D \mid L) \times p(B \mapsto_z D)$$

Like Becker and Gouskova (2016), I take the grammar to be deterministic: $p(B \mapsto_z D) \in \{0, 1\}$. However, the model is fully compatible with a probabilistic phonological grammar.

2.2 A segment exceptionality model

For Russian vowel-zero alternation, the alternative to the morpheme exceptionality view is a distinction between two types of underlying vowels, which is neutralized on the surface: either a feature distinction (Lightner 1965; Halle and Matushansky 2006) or a difference in autosegmental structure (Kenstowicz and Rubach 1987; Scheer 2019 i.a.). To be non-committal, I assume the distinction between vowels to be encoded by the feature F.

The usual analysis holds that a vowel marked with F only surfaces when preceding a syllable with a vowel marked with F (Scheer 2011). The form *mox* is *mo$_F$xo$_F$* underlyingly and the form *mx-a* is *mo$_F$xa*. Figure 2 shows the application of rules in (1).

(1) a. $V_F \rightarrow V$ /___C_0V_F

b. $V_F \rightarrow \emptyset$

mo$_F$x-o$_F$	mo$_F$x-a	nos-o$_F$	nos-a	UR
moxo$_F$	mo$_F$xa	noso$_F$	nosa	Rule (1a)
mox	mxa	nos	nosa	Rule (1b)

Figure 2: Derivation of the alternation.

Under the segment exceptionality view, extending an alternation to a nonce word is a matter of assigning an appropriate UR: i.e., accepting the nonce UR with a F-marked vowel (see Nelson 2024:61 for a similar statement). Gradient nonce word results will be a consequence of gradient similarity of a nonce UR to existing URs.

I propose that similarity of a nonce UR is determined by the speakers' knowledge of possible sequences of underlying segments in their lexicon: call it morpheme structure phonotactics, a generalization of morpheme structure constraints (see the overview in Booij 2011). One is more likely to accept a nonce item alternation if the required underlying form conforms to the distribution of underlying segments in their lexicon (see also Hypothesis 4 of Ernestus and Baayen 2003).

$$p(nom \sim gen_\emptyset) \propto s(nom_F \mid U) s(gen_\emptyset \mid L)$$

How to read the proportion: the probability $p(nom \sim gen_\emptyset)$ of accepting a derived form with vowel deletion is proportional to the product of $s(gen_\emptyset \mid L)$, the measure of phonotactic similarity of the derived form $gen_\emptyset$ to the lexicon L and $s(nom_F \mid U)$, the measure of phonotactic similarity of the UR nom_F to the lexicon of URs U.

To generalize, $p(B \sim D)$, the probability of the alternation $B \sim D$ between the base B and the derived form D, is proportional to: $s(D \mid L)$, the measure of phonotactic similarity of D to the lexicon L, $s(B_u \mid U)$, the measure of phonotactic similarity of the UR B_u to the lexicon of URs U, and $p(B_u \mapsto D)$, the probability of the $B_u \mapsto D$ mapping.

$$p(B \sim D) \propto s(B_u \mid U) \times s(D \mid L) \times p(B_u \mapsto D)$$

The assumption, again, is that the grammar has a single output: $p(B_U \mapsto D) \in \{0, 1\}$. However, there is no principled conflict with a probabilistic phonological grammar.

2.3 How either model captures tendencies

Let us consider how either model accounts for the phonotactic tendencies in alternating morphemes, such as the *CC# ban (Section 2.1; Gouskova and Becker 2013). To repeat: even though 17% of Russian masculine nouns end in a consonant cluster, no masculine noun undergoing the vowel-zero alternation does.

Any model based on morpheme exceptionality has no trouble capturing the generalization: if the sublexicon of alternating words has no CC-final ones, that's a strong phonotactic restriction which a nonce item is unlikely to overcome (as results of Gouskova and Becker 2013 show).

Things seem harder for a segment exceptionality model: assuming that only 1-, 2-, and 3-grams matter for phonotactics (as held by Hayes and Wilson 2008; Wilson and Gallagher 2018; Gouskova and Gallagher 2020 i.a.), the ban on a CC# sequences cannot be directly expressed with a reference to the alternating vowel. However, not all phonotactic restrictions have to be captured in a direct way.

An independent fact is that only the last vowels of morphemes may alternate (Gouskova 2012). Therefore, V_FCC can only occur word-finally, in a V_FCC# environment. A ban on *CC# items that undergo the alternation falls out of a ban on an underlying V_FCC sequences together with the generalization that V_F must be last vowel in the morpheme (which is a consequence of alternation-driven learning: a V_F that isn't the last vowel of the morpheme either always surfaces as V or zero; Scheer 2019:212 makes a similar point in his discussion of Gouskova and Becker's findings).

To capture the tendencies, both model rely on an independent measure of phonotactic similarity. The next section discusses two such models.

3 The phonotactic models

There is no shortage of phonotactic models in the literature (see Mayer 2025 for a recent discussion). For the purposes of this paper, I contrast two: the gradient model of Hayes and Wilson (2008) and the categorical model of Durvasula (2026). See Durvasula's paper for a thorough discussion of the role of gradience in phonotactic modelling.

A phonotactic model is characterized by two core properties. The first is the *hypothesis space*: what kinds of generalizations can be extracted from the lexicon. Hayes and Wilson's (2008) model is able to formulate constraints on n-grams of bundles of features. For example, $*\langle \begin{bmatrix} +son \\ -lab \end{bmatrix}, \begin{bmatrix} -son \\ -lab \end{bmatrix} \rangle$ which penalizes non-labial sonorant-obstruent sequences (*nk, nd,...) is a possible constraint. On the other hand, the model of Durvasula (2026) restricts the hypothesis space to segmental n-grams and n-grams of valued features: so, $*\langle [+son], [-son] \rangle$ is a possible constraint, but $*\langle \begin{bmatrix} +son \\ -syll \end{bmatrix}, \begin{bmatrix} +son \\ +syll \end{bmatrix} \rangle$ is not, since it refers to bundles of valued features. It is possible to restrict the hypothesis space even further and only entertain segment-based n-grams. Appendix A discusses two such models.

The second core property is *gradience* of evaluation. In the model of Hayes and Wilson (2008), constraints (illicit n-grams) are unequal in their contribution to the resulting score. The harmony score is the number of times the word violates a constraint (C_n) times its weight (w_n, summed over N constraints. The constraints and their weights are determined using a MaxEnt learning mechanism, described in ibid. Hence, I refer to their model as a MaxEnt model.

$$h(x) = \Sigma_{n=1}^{N} w_n C_n(x)$$

The model of Durvasula (2026), however, treats all constraints as equally weighted. So the output of the model is the count of occurrences of a banned n-gram in the word $C_n(x)$, summed over all banned n-grams. Therefore, the weights are all equal whereas the constraints themselves are simply those in the hypothesis space which are absent from the corpus. Hence, I refer to his model as a segmental-featural cost (SF cost) model. Appendix B discusses a modification of Durvasula's model without segmental n-grams.

$$m(x) = \Sigma_{n=1}^{N} C_n(x)$$

In this work, I report the results of the segmental-featural cost models over 2- and 3-grams (SF 2,3-cost). The score of the segmental-featural cost models is transformed into probabilities of accepting / rejecting any given nonce item by taking $\frac{1}{m(x)+1}$ ($= e^{-ln(x+1)}$; a phonotactically ideal nonce item will always be accepted by participants, and so on). A probability estimate for the MaxEnt model is taken from Becker and Gouskova (2016).

Before we turn to the evaluation of the models, I wish to emphasize that the categorical / gradient divide concerns degrees of sequence illicitness (weighting). Any linking hypothesis between a phonotactic score and a nonce word task is a way to transform the score into a probability of accepting a given nonce item—so the fact that both a MaxEnt-based model and cost-based models ultimately return a value in the range from 0 to 1 should not obscure the different assumptions about weighting, and, thus, the relevance of the findings to the gradience debate in phonotactic modelling.

4 Model evaluation

This section compares the sublexical MaxEnt model of Becker and Gouskova (2016) to a morpheme exceptionality based model (the sublexical model) using an SF cost phonotactic model (to see whether the sublexical model requires any strong committments to a phonotactic model) and to a segment exceptionality based model using an SF cost phonotactic model (to see whether the morpheme structure phonotactics proposal is viable). I do not report a morpheme structure phonotactics model based on MaxEnt phonotactics due to its lack of theoretical interest: a proponent of MaxEnt grammars is unlikely to be interested in segment exceptionality and vice versa.

The corpus materials, experimental data, feature decompositions, and simulations for the MaxEnt sublexical model are taken from Becker and Gouskova and are available at https://becker.phonologist.org/projects/yers/. The materials for my own simulations are available at https://github.com/antidanyar/msp.

I first present quantitative evaluation of the models (Sections 4.3-4.4), which slightly favors the morpheme structure phonotactics model, and then discuss how the model captures the generalizations about alternating morphemes covered by Gouskova and Becker (2013), explaining its quantitative success (Section 4.5).

4.1 The corpus data & features

The Russian lexicon is approximated by the list of 2nd declension masculine nominals in their NOM.SG and GEN.SG forms, with annotation regarding vowel-zero alternations (adapted from Zaliznjak's 1977 dictionary of Russian). For example, the data for the word *šov* 'seam' involve the forms *šov* and *šv-a* (NOM.SG and GEN.SG) with an indication that it is a word that undergoes the vowel-zero alternation. An important caveat is that some words in the training data are multi-morphemic and contain derivational affixes: for example, the word *ščen-ok* $\sim$ *ščen-k-a* 'puppy' has the diminutive affix *-(o)k* which undergoes the alternation. As far as I am aware, this simplification does not jeopardize any claims made in the paper—I leave replication on a more realistic lexicon of URs for future work.

To get the valued feature n-grams required for the segment-featural cost models, I have used the feature decomposition employed by Becker and Gouskova (2016). To compute the phonotactic scores over URs, conservative underlying representations have been assumed, which correspond to the dictionary entries of NOM.SG with one change: vowels that undergo the vowel-zero alternation are marked as o_F and e_F to make them distinct from vowels that do not.

For the segmental-featural cost models, I have supplemented the feature decomposition with a $\pm F$ feature: o_F and e_F are $+F$ (since only /o,e/ alternate) and all other segments are $-F$. That choice did not end up being substantive: see Table 16 in Appendix for the feature matrix employed in the simulations reported here & Appendix B for a discussion of results with alternative feature sets.

4.2 The experimental data

Participants were shown a nonce item and asked to rate it on a 1-5 scale. They were then given two suffixed forms, with deletion and without, and were asked if they are acceptable: see Figure 3 for an example trial. The nonce items varied in the quality of the alternating vowel (*o* or *e*) number of syllables (monosyllabic vs. disyllabic where disyllabics have a CV first syllable), complexity of onsets/codas (CVC, CCVC, CVCC) and the sonority sequence of the variant without vowel (TR, TRT, and so on).

Becker and Gouskova (2016) note that the reported wellformedness of the base form didn't fully determine the acceptability of the derived forms, I employ those estimates as baseline model

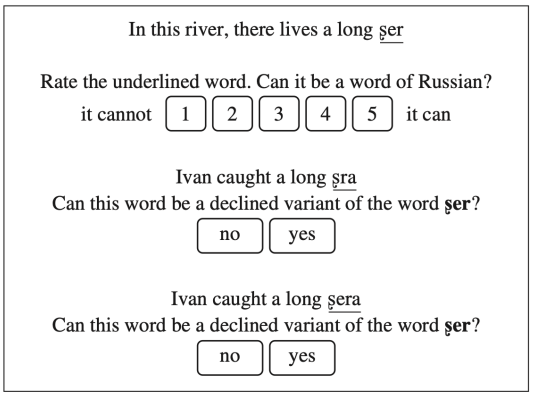


Figure 3: Translation of an example trial showing a nominative base and two genitive derivatives. Reproduced from Becker and Gouskova (2016), p.398.

(by taking $\frac{1}{6-x}$)–if anything, a successful morpho-phonological model must outperform the effect of the phonotactic wellformedness of the base reported by the participant.

4.3 Evaluation I: aggregates over types

Following Becker and Gouskova (2016), I evaluate the models based on the average score given to each syllabic type (their design resulted in most items only being evaluated by a single participant, see Section 3.3 of ibid.). Only aggregation by syllabic types makes it possible for correlation scores to be informative, given the binary choice task.

Table 1 presents three correlation metrics with bootstrapped 95% confidence intervals (see Albright 2009 and Durvasula 2026 for some discussion of the use of correlation metrics for evaluation of phonotactic models).

Model	Pearson r [95% CI]	Spearman ρ [95% CI]	Kendall τ_b [95% CI]
MaxEnt (morpheme)	**.893** [.837, .939]	.854 [.746, .915]	.673 [.566, .769]
SF 2,3-cost (segment)	.844 [.778, .925]	**.858** [.729, .928]	**.683** [.556, .796]
Baseline	.738 [.640, .825]	.655 [.480, .758]	.432 [.293, .560]
SF 2,3-cost (morpheme)	.713 [.605, .828]	.804 [.663, .885]	.607 [.478, .727]
SF 2-cost (morpheme)	.654 [.504, .777]	.609 [.390, .754]	.411 [.240, .561]
SF 2-cost (segment)	.552 [.344, .718]	.519 [.253, .716]	.373 [.184, .529]

Table 1: Syllabic type-level correlations between model predictions and empirical acceptability estimates. Brackets indicate 95% confidence intervals obtained by bootstrap resampling ($n = 5000$).

The aggregate correlation values suggest that the two best performing models are the sublexical MaxEnt model of Becker and Gouskova (2016) and the morpheme structure phonotactics model (SF-cost segment) over 2- and 3-grams. Based on the 95% CIs, these models are the only ones that significantly outperform the baseline.

However, no choice between the top two models is possible on the basis of the correlation metrics and their CIs, suggesting that another metric is needed. Additionally, one might be uncomfortable with evaluating model fit against averages over syllabic types—I now present a metric computed over raw experimental data that will establish a slight preference towards the morpheme structure phonotactics model.

4.4 Evaluation II: AUC score

The binary choice task of Becker and Gouskova (2016) motivates model evaluation based on the proportion of cross-class observation pairs they get correctly: if item x is accepted by one participant and item y is rejected by another participant, x should be ranked as more word-like than y.

Here, I present weighted AUC score computed by assigning each observation a weight $w_i = \frac{1}{N_i}$ where N_i is a number of observations for i-th nonce item (every nonce word contributes an equal mass of 1). The function $S(x, y)$ returns 1 if $f(x) > f(y)$, .5 if $f(x) = f(y)$, and 0 if $f(x) < f(y)$.

$$AUC = \frac{\sum_{x\in\mathcal{D}^+}\sum_{y\in\mathcal{D}^-} w_x w_y S(x,y)}{\sum_{x\in\mathcal{D}^+} w_x \sum_{y\in\mathcal{D}^-} w_y}$$

Table 2 suggests that the segment-based morpho-phonological model based on segmental-featural 2,3-cost phonotactic score outperforms alternatives. However, 95% CIs themselves do not allow to establish significant model differences. Similarly, the ROC AUC curve in Figure 4 does not show a clear superior model.

Model	AUC-ROC (item-weighted)
SF 2,3-cost (segment)	**.652** [.629, .673]
SF 2,3-cost (morpheme)	.640 [.617, .661]
MaxEnt (morpheme)	.637 [.615, .659]
Baseline	.613 [.592, .634]

Table 2: Item-normalized AUC score (larger is better, .5 is as good as chance). Brackets indicate 95% confidence intervals obtained by bootstrap resampling ($n = 5000$). Only models that outperform baseline are reported.

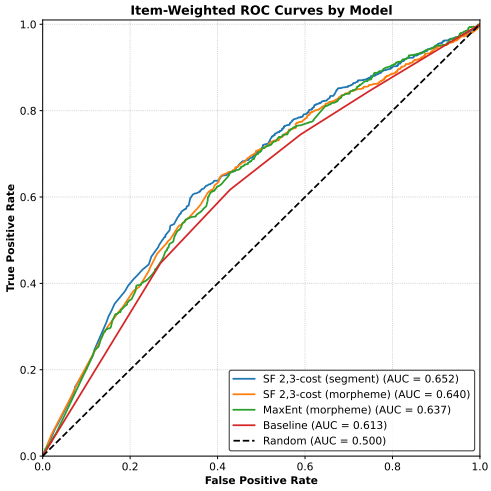


Figure 4: Item-weighted AUC-ROC curves by model.

To help distinguish between models, Table 3 presents bootstrapped differences in AUC score for the top model in Table 2 (namely, SF 2,3-cost; the implementation of the morpheme structure phonotactics model) and the rest of the models.

The bootstrapped differences suggest that the SF 2,3-cost model outperforms the alternatives: albeit the lower bound of the interval of difference between the MaxEnt model and the segmental SF 2,3-cost model is extremely close to zero.

Compared model	Mean Δ AUC [95% CI]
SF 2,3-cost (morpheme)	.012 [.006, .017]
MaxEnt (morpheme)	.014 [.002, .026]
Baseline	.039 [.021, .056]

Table 3: Pairwise differences in AUC scores between the segment-based segment-featural 2,3-cost model and other models. Values represent the mean difference (ΔAUC) with 95% confidence intervals in brackets, obtained via grouped bootstrap resampling ($n = 5000$). Intervals that do not include zero indicate a statistically significant difference at $\alpha = .05$.

Both evaluation metrics thus suggest that the segment-based morpho-phonological model, supplemented with an appropriate phonotactic score (the segmental-featural model of Durvasula 2026), predicts the productivity of Russian vowel-zero alternations at least just as well as the model presented by Becker and Gouskova (2016). Pairwise AUC differences provide a promising result in favor of the morpheme structure phonotactics model—but I leave the matter unsettled here.

4.5 Evaluation III: phonotactic tendencies

The morpheme structure phonotactics model (SF 2,3-cost) is able to account for the gradient productivity of Russian vowel-zero alternations. It's worth investigating how. In this section, I present observations regarding the behavior of the model with respect to the phonotactic tendencies discovered by Gouskova and Becker (2013).

4.5.1 Complex coda restriction

Nonce words that end in a CVCC syllable are less likely to undergo the vowel-zero alternation than nonce words that end in a CVC or CCVC syllable, reflecting the fact that while 17% of Russian masculine nouns end in CC# sequence, no alternating masculine nouns do.

For example, the nonce word *postor* is more likely to have the form *postr-a* than the nonce word *posotr* is to have the form *postr-a*. The model supports this: the former has one illicit UR n-gram (namely, /to$_F$r/) while the latter has two: /so$_F$t, o$_F$tr/.

Nonce word	Illicit segment 2,3-grams	SF 2,3-Cost
poso$_F$tr	/so$_F$t, o$_F$tr/	242
posto$_F$r	/to$_F$r/	1

Table 4: Illicit segment 2,3-grams and SF 2,3-cost for the nonce URs.

Note, however, that the complex coda UR *poso$_F$tr* violates 242 segment & feature n-grams while *posto$_F$r* only violates 1. The reason is due to the non-accidental complex coda ban in the lexicon. While 2 violated n-grams are segmental, all others are valued feature n-grams: the gap is natural class-based.

Nonce word	Illicit feature 2,3-grams	SF 2,3-Cost
poso$_F$tr	/+flt, −$syll$,−$syll$/ /+flt, −son,+son/ ...	242
posto$_F$r	—	1

Table 5: Illicit valued feature 2,3-grams and SF 2,3-cost for the nonce URs.

Consider, for example, that the nonce word *poso$_F$tr* includes an illicit valued feature n-grams /+flt, −$syll$,−$syll$/ and /+flt, −son,+son/: it suggests that there are no V$_F$CC sequences in the UR lexicon. As discussed in Section 2.3, this observation derives the complex coda restriction since all alternating vowels are the last vowels in their respective morphemes.

4.5.2 Sonority restrictions

Then, consider the sonority restrictions. nonce words that end in a TRVT syllable are less likely to undergo nonce words that end in a TTVR syllable. This reflects the fact that while 20% of Russian masculine nouns ends in a CR(V)C sequence, only .3% of alternating ones do.

For example, the nonce word *postor* is more likely to have the form *postr-a* than *posrot* is to have the form *posrt-a*. The nonce word *postor*, again, has as single illicit segmental n-gram, while *posrot* has two (although the absence of the /osr/ trigram is likely accidental and due to corpus).

Nonce word	Illicit segment 2,3-grams	SF 2,3-Cost
posro$_F$t	/osr, sro$_F$/	16
posto$_F$r	/to$_F$r/	1

Table 6: Illicit segment 2,3-grams and SF 2,3-cost for the nonce URs.

Importantly, the 14 illicit valued feature n-grams are all due to the /sro$_F$/ segment trigram. They are not the /$+cons$, $+son$, $+flt$/ trigram: since the gap isn't complete.

Nonce word	Illicit feature 2,3-grams	SF 2,3-Cost
posro$_F$t	/$-voi$, $+son$,$+flt$/ /$+strid$, $+son$,$+flt$/	16
	...	
posto$_F$r	—	1

Table 7: Illicit valued feature 2,3-grams and SF 2,3-cost for the nonce URs.

However, there are plenty of subcategories of consonants that do not occur in native CRV$_F$ trigrams: for example, voiceless obstruents or stridents, providing a natural class-based reason for the model to disfavor nonce items that end in CRVC syllables as a group.

4.5.3 Monosyllabicity

The final restriction is monosyllabicity: monosyllabic items are less likely to undergo the alternation. This reflects the observation that while 8% of Russian masculine nouns are monosyllabic, only .7% of alternating ones are.

For example, the nonce word *pokor* is more likely to have the form *pokr-a* than the nonce word *kor* is to have the form *kr-a*. This is reflected in violated segmental n-grams: the nonce UR *poko$_F$r* has none, while the nonce UR *ko$_F$r* has one.

Nonce word	Illicit segment 2,3-grams	SF 2,3-Cost
ko$_F$r	/#ko$_F$/	4
poko$_F$r	—	0

Table 8: Illicit segment 2,3-grams and SF 2,3-cost for the nonce URs.

And again, the segmental trigram /#ko$_F$/ instantiates multiple natural class-based gaps: for example, all /#, $+dors$, $+flt$/ trigrams are absent from the UR lexicon—meaning that no monosyllabic alternating masculine nouns start with dorsals.

Nonce word	Illicit feature 2,3-grams	SF 2,3-Cost
ko$_F$r	/#, $+dors$, $+flt$/ /# $+high$,$+flt$/	4
	...	
poko$_F$r	—	0

Table 9: Illicit valued feature 2,3-grams and SF 2,3-cost for the nonce URs.

Gaps in boundary initial #CV$_F$ trigrams are an approximation of the restrictions on monosyllabic alternating nouns—since V$_F$ must be the last vowel in the morpheme, words with a #CV$_F$ trigram are monosyllabic alternating words. The relevant pattern in the data is that only 10 segmental #CV$_F$ sequences are found in the UR lexicon (whereas there are 64 with non-floating mid vowels). Therefore, some monosyllabic nonce items will be penalized for #CV$_F$ sequences and sequences of valued features they exemplify, leading to a lower acceptability of monosyllabic alternating nonce words.

4.5.4 Double-checking the effect of tendencies

To evaluate the effect of phonotactic tendencies, I ran Ordinary Least Squares (OLS) linear regression with the following structure. The model output for the nonce words served as the dependent variable, while the predictors included syllable count, complex coda, and sonority profile (whether the output of vowel deletion is TR, TTR, or RTR).

The predictors also included the number of consonants in the output of vowel deletion and the vowel itself, combined with an interaction factor between sonority sequencing and number of consonants (following the regression model reported by Becker and Gouskova 2016:404). To safeguard against potential heteroskedasticity and non-normality in the model residuals, all parameter estimates were calculated using Heteroskedasticity-Consistent robust standard errors (HC3).

The results are reported in Table 10. The main conclusion is that all three phonotactic tendencies are significant predictors of the output of the proposed model (boldfaced).

Predictor	β	SE	z	p
Intercept	.167	.006	28.56	$< .001$
Monosyllabic	$-.018$	.003	-6.39	$< .001$
Complex coda	$-.030$	.002	-14.89	$< .001$
Vowel	$-.030$	.003	-8.87	$< .001$
Sonority profile	.037	.011	3.43	.001
Number of Cs	$-.132$	.005	-27.09	$< .001$
SP $\times$ # of Cs	$-.019$	.011	-1.74	.082

Table 10: OLS regression results probing the effect of phonotactic tendencies on the outputs of the morpheme structure phonotactics model.

The estimates show that the directionality of effects are correct—monosyllables and complex codas bring probability estimated by the proposed model down, whereas TTR and RTR outputs of vowel deletion raise it. I therefore conclude that the morpheme structure phonotactics model succeeds in capturing the effect of phonotactic tendencies on productivity of Russian vowel-zero alternations discovered by Gouskova and Becker (2013) and Becker and Gouskova (2016), along the lines discussed in Sections 4.5.1–4.5.3.

5 Conclusion & outlook

Nonce word tasks require careful consideration of the linking hypotheses between linguistic knowledge and performance in experimental settings. In this paper, I have pursued an explicit linking hypothesis between morpho-phonological nonce word tasks and "abstract phonology" approaches to morpho-phonological phenomena which rely on covert differences in underlying representations to achieve marking of phonological segments themselves as targets of exceptional phonological processes.

The core idea of the proposed morpheme structure phonotactics model is phonotactic evaluation of underlying representations (extending the ideas put forth by Ernestus and Baayen 2003 to abstract underlying segments). On top of that, the phonotactic evaluation itself is done using an n-gram counting model—suggesting that a weighted constraints model is not required for phonotactic wellformedness effects (Gorman 2013; Kostyszyn and Heinz 2021; Durvasula 2026).

The main result of this study is that the morpheme structure phonotactics model performs just as well as the sublexical MaxEnt model of Becker and Gouskova (2016)—and, perhaps, exceeds it—on their own nonce word judgement data regarding productivity of Russian vowel-zero alternations. The success isn't merely quantitative: as shown in Section 4.5, the phonotactic tendencies that characterize the masculine nouns that undergo the alternation are captured by a model which is only sensitive to the distribution of exceptional segments in the existing URs.

I believe the linking hypothesis to have two major conceptual predecessors in the literature: the traditional notion of morpheme structure constraints, negative statements about distribution of segments and their sequences in URs (Chomsky and Halle 1968; see the overview in Booij 2011 and see the recent defenses in Rasin and Katzir 2020; Gouskova 2023; Rasin 2025) and the strand conceptual criticism of phonotactic and morpho-phonological nonce word tasks, which takes the results to reflect static trends in the lexicon which are grammatically inert (Schütze 2005; Gorman 2013; Scheer 2019).

Whether this perspective extends to other alternations and languages remains a question for future work, but the present case study shows that morpho-phonological nonce word results are compatible with a wider range of phonological theories than previously thought. The findings thus neutralize the apparent conflict between morpho-phonological nonce word results and the empirical necessity of treating phonological segments themselves as targets or triggers of morpho-phonological processes (Martínez 2008; Rubach 2013, 2016; Round 2017; Zimmermann 2025 *i.a.*).

Acknowledgments

I thank Maria Gouskova, Kaustubh Ghoshal, Gillian Gallagher, Kyle Gorman, Adam Albright, and two anonymous reviewers for helpful feedback.

References

Adam Albright. 2009. Feature-based generalisation as a source of gradient acceptability. *Phonology*, 26(1):9–41.

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

Blake Allen and Michael Becker. 2015. Learning alternations from surface forms with sublexical phonology. *Unpublished manuscript, University of British*

Columbia and Stony Brook University. Available as lingbuzz/002503.

Michael Becker and Maria Gouskova. 2016. Source-oriented generalizations as grammar inference in russian vowel deletion. *Linguistic inquiry*, 47(3):391–425.

Michael Becker, Nihan Ketrez, and Andrew Nevins. 2011. The surfeit of the stimulus: Analytic biases filter lexical statistics in turkish laryngeal alternations. *Language*, 87(1):84–125.

Geert Booij. 2011. Morpheme structure constraints. *The Blackwell companion to phonology*, pages 1–21.

Noam Chomsky and Morris Halle. 1968. *The sound pattern of English*. Harper&Row.

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.

Karthik Durvasula. 2026. A closer look at what/how we can learn from computational modelling of phonotactics. *Ms. University of Michigan*.

Mirjam Ernestus and R Harald Baayen. 2003. Predicting the unpredictable: Interpreting neutralized segments in Dutch. *Language*, 79(1):5–38.

Kyle Gorman. 2013. *Generative phonotactics*. Ph.D. thesis, University of Pennsylvania.

Maria Gouskova. 2012. Unexceptional segments. *Natural Language & Linguistic Theory*, 30(1):79–133.

Maria Gouskova. 2023. Mscs in positional neutralisation: the problem of gapped inventories. *Phonology*, 40(3-4):229–265.

Maria Gouskova and Michael Becker. 2013. Nonce words show that Russian yer alternations are governed by the grammar. *Natural Language & Linguistic Theory*, 31(3):735–765.

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

Morris Halle. 1959. *The Sound Pattern of Russian. A Linguistic and Acoustical Investigation*. 's-Gravenhage.

Morris Halle and Ora Matushansky. 2006. The morphophonology of russian adjectival inflection. *Linguistic inquiry*, 37(3):351–404.

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

Bruce Hayes, Kie Zuraw, Péter Siptár, and Zsuzsa Londe. 2009. Natural and unnatural constraints in hungarian vowel harmony. *Language*, 85(4):822–863.

Michael Kenstowicz and Jerzy Rubach. 1987. The phonology of syllabic nuclei in slovak. *Language*, pages 463–497.

Kalina Kostyszyn and Jeffrey Heinz. 2021. Categorical account of gradient acceptability of word-initial polish onsets. In *Proceedings of the Annual Meetings on Phonology*.

Theodore McGraw Lightner. 1965. *Segmental phonology of modern standard Russian.* Ph.D. thesis, Massachusetts Institute of Technology.

Michal Temkin Martínez. 2008. Exceptionality and variation in modern hebrew spirantization. In *WCCFL 27*, pages 413–421.

Connor Mayer. 2025. Reconciling categorical and gradient models of phonotactics. *Society for Computation in Linguistics*, 8(1).

Connor Mayer and Max Nelson. 2020. Phonotactic learning with neural language models. *Society for Computation in Linguistics*, 3(1).

Scott Nelson. 2024. *The Computational Structure of Phonological and Phonetic Knowledge*. Ph.D. thesis, State University of New York at Stony Brook.

Joe Pater. 2008. Morpheme-specific phonology: Constraint indexation and inconsistency resolution. In *Phonological argumentation: Essays on evidence and motivation*. Equinox Publishing Ltd.

Ezer Rasin. 2025. Morpheme structure constraints solve three puzzles for theories of blocking in nonderived environments. *Linguistic Inquiry*, 56(4):713–749.

Ezer Rasin and Roni Katzir. 2020. A conditional learnability argument for constraints on underlying representations. *Journal of Linguistics*, 56(4):745–773.

Erich Round. 2017. Phonological exceptionality is localized to phonological elements: The argument from learnability and yidiny word-final deletion. *On looking into words (and beyond): Structures, relations, analyses*, pages 59–97.

Jerzy Rubach. 2013. Exceptional segments in polish. *Natural Language & Linguistic Theory*, 31(4):1139–1162.

Jerzy Rubach. 2016. Polish yers: Representation and analysis1. *Journal of linguistics*, 52(2):421–466.

Tobias Scheer. 2011. Slavic yers. In *The Blackwell companion to phonology*, pages 1–27. Wiley Online Library.

Tobias Scheer. 2019. On the difference between the lexicon and computation (regarding slavic yers). *Linguistic inquiry*, 50(1):197–218.

Carson T Schütze. 2005. Thinking about what we are asking speakers to do. *Linguistic evidence: Empirical, theoretical, and computational perspectives*, pages 457–485.

Colin Wilson and Gillian Gallagher. 2018. Accidental gaps and surface-based phonotactic learning: A case study of south bolivian quechua. *Linguistic Inquiry*, 49(3):610–623.

Andrej Anatoljevich Zaliznjak. 1977. *Grammatičeskij slovar' russkogo jazyka [A grammatical dictionary of the Russian language]*. Moscow: Russkij Jazyk.

Eva Zimmermann. 2025. Cooperation at the morpho-phonology interface: An argument for phonological locality. In *Proceedings of the 41st West Coast Conference on Formal Linguistics*, pages 18–38.

Eva Zimmermann. 2026. Morphemes in phonology. In Paul de Lacy and Adam Jardine, editors, *The Cambridge Handbook of Phonology (2nd ed.)*. Cambridge University Press.

A Alternative phonotactic models

A.1 Accidental gaps

As mentioned earlier, Durvasula's (2026) model is not the only categorical phonotactic model on the market. There are arguably even simpler alternatives. For example, Gorman (2013) proposes a Boolean model that maps a nonce word to 0 if any of its n-grams are unattested in the corpus. Kostyszyn and Heinz (2021) propose a segmental cost model that maps a nonce word to how many of its n-grams are unattested in the corpus.

Both alternatives are subject to the *accidental gap* argument of Wilson and Gallagher (2018)—they are unable to distinguish between a genuine, natural-class based gap in the data, and an accidental gap. An illustration might help. Consider a corpus where voiced stops do not occur word-finally (*[$+voi$]#) and the segment /p/ does not occur non-finally.

Sequence	Boolean	S-Cost	S&F-Cost
at#	1	0	0
ap#	0	1	1
ad#	0	1	2
ab#	0	1	2

Table 11: Different scores in a toy example.

As discussed in Wilson and Gallagher (2018) for Quechua, a phonotactic model must distinguish between an accidental gap like **p#* and a real, natural class-based gap of **d#* and **b#*. Since both Boolean and segmental cost models only take attested segmental sequences into account, the difference is lost—as indicated by the scores. Only the segment-feature cost model is able to make the right cut: absence of the ⟨[$+voi$],#⟩ bigram adds onto the score of **d#* and **b#*. The discussion in Section 4.5.1. presents additional evidence in favor of sensitivity to natural class-based gaps.

The inherent issues of the alternative categorical models set in place suggest their inferior performance relative to both Becker&Gouskova's MaxEnt model and the morpheme structure phonotactics model supplemented with Durvasula's phonotactic scoring. As the next section shows, that is indeed the case.

A.2 Evaluation of alternative models

Here, I report the correlation metrics on all of the models: the numerical prefix refers to whether the model is sensitive to unattested bi- and tri-grams, while S cost refers to the segmental cost model of Kostyszyn and Heinz (2021) and Boolean refers to the Boolean model of Gorman (2013).

Model	Pearson r [95% CI]	Spearman ρ [95% CI]	Kendall τ_b [95% CI]
MaxEnt (morpheme)	**.893** [.837, .939]	.854 [.746, .915]	.673 [.566, .769]
SF 2,3-cost (segment)	.844 [.778, .925]	**.858** [.729, .928]	**.683** [.556, .796]
S 2,3-cost (segment)	.781 [.694, .874]	.792 [.630, .882]	.609 [.469, .724]
Baseline	.738 [.640, .825]	.655 [.480, .758]	.432 [.293, .560]
SF 2,3-cost (morpheme)	.713 [.605, .828]	.804 [.663, .885]	.607 [.478, .727]
S 2-cost (morpheme)	.668 [.511, .786]	.620 [.380, .762]	.425 [.243, .571]
S 2,3-cost (morpheme)	.656 [.534, .788]	.708 [.525, .815]	.500 [.348, .631]
2-Boolean (morpheme)	.656 [.495, .781]	.591 [.357, .741]	.402 [.218, .559]
3-Boolean (morpheme)	.656 [.493, .778]	.591 [.359, .742]	.402 [.219, .548]
SF 2-cost (morpheme)	.654 [.504, .777]	.609 [.390, .754]	.411 [.240, .561]
SF 2-cost (segment)	.552 [.344, .718]	.519 [.253, .716]	.373 [.184, .529]
S 2-cost (segment)	.357 [.059, .604]	.359 [.054, .606]	.250 [.041, .440]
2-Boolean (segment)	.273 [-.029, .546]	.286 [-.027, .565]	.214 [-.014, .431]
3-Boolean (segment)	.273 [-.029, .554]	.286 [-.037, .579]	.214 [-.023, .436]

Table 12: Syllabic type-level correlations between model predictions and empirical acceptability estimates. Brackets indicate 95% confidence intervals obtained by bootstrap resampling ($n = 5000$).

No alternative model outperforms the baseline (which is based on the mean reported wellformedness of the nominative bases in the study). For completeness' sake, I will also report the AUC scores

of the S 2,3-cost (segment) model relative to the baseline, SF 2,3-cost, and MaxEnt model: the S 2,3-cost (segment) model is outpeformed by both SF 2,3-cost, and MaxEnt models. I conclude that natural class sensitivity is required for modelling the effects found in productivity of Russian vowel-zero alternations.

Model	AUC-ROC (item-weighted)
SF 2,3-cost (segment)	**.652** [.629, .673]
MaxEnt (morpheme)	.637 [.615, .659]
Baseline	.613 [.592, .634]
S 2,3-cost (segment)	.611 [.588, .632]

Table 13: Model comparison based on item-normalized AUC score (larger is better, .5 is as good as chance). Brackets indicated 95% confidence intervals obtained by bootstrap resampling ($n = 5000$).

B On feature decompositions

The feature decomposition employed in this paper is shown in Table 16 (next page). The core decision was to treat the $\pm float$ contrast as privative. This appendix describes the results of simulations with other feature decompositions, focusing on the AUC score of the SF 2,3-cost model (Binary-Flt). I entertain three additional possibilities: treating *float* as privative (only V_F is marked as $+$: Priv-Flt-Plus), treating *float* as privative (only regular segments are marked as $-$: Priv-Flt-Minus), and dispensing with *float*, marking V_F as underspecified for syllabicity (Priv-Syll).

Model	AUC-ROC (item-weighted)
Binary-Flt (SF 2,3-cost)	.652 [.629, .673]
Priv-Flt-Plus	.652 [.629, .673]
Priv-Flt-Minus	.643 [.620, .665]
Priv-Syll	.643 [.620, .664]

Table 14: Comparison of SF 2,3-cost models with different feature matrices based on item-normalized AUC score. Brackets indicated 95% confidence intervals obtained by bootstrap resampling ($n = 5000$).

The conclusion is that the privative / binary choice has no effect. What does have an effect is relative *markedness* of the vowel undergoing the alternation. If there is no feature that distinguishes it from the rest—model performance drops.

In that light, another worthwhile comparison is to see how big is the effect of the *segmental* part of the model—since the UR segment does provide another way to distinguish undergoing vowels from the rest. I report AUC scores where both UR and SR wellformedness is only evaluated relative to valued feature n-grams, for all four feature decompositions (thus, they are F 2,3-cost models).

Model	AUC-ROC (item-weighted)
Binary-Flt (F 2,3-cost)	.648 [.627, .670]
Priv-Flt-Plus	.648 [.627, .670]
Priv-Flt-Minus	.615 [.595, .635]
Priv-Syll	.615 [.596, .634]

Table 15: Comparison of F 2,3-cost models with different feature matrices based on item-normalized AUC score. Brackets indicated 95% confidence intervals obtained by bootstrap resampling ($n = 5000$).

Interestingly, omitting segment identity does not lead to a major drop in model performance. I leave further exploration of this finding for further work—I only note here that whatever the feature decomposition, it must be able to provide a natural class that only includes the vowels that undergo vowel deletion.

Segment	syll	cons	approx	son	cont	delrel	lateral	nasal	voice	dor	lab	cor	strid	ant	retro	high	low	back	stress	float	boundary
p	–	+	–	–	–	–	–	–	–	0	+	0	0	0	0	0	0	+	0	–	–
pʲ	–	+	–	–	–	–	–	–	–	0	+	0	0	0	0	0	0	–	0	–	–
b	–	+	–	–	–	–	–	–	+	0	+	0	0	0	0	0	0	+	0	–	–
bʲ	–	+	–	–	–	–	–	–	+	0	+	0	0	0	0	0	0	–	0	–	–
m	–	+	–	+	–	0	–	+	+	0	+	0	0	0	0	0	0	+	0	–	–
mʲ	–	+	–	+	–	0	–	+	+	0	+	0	0	0	0	0	0	–	0	–	–
f	–	+	–	–	+	+	–	–	–	0	+	0	0	0	0	0	0	+	0	–	–
fʲ	–	+	–	–	+	+	–	–	–	0	+	0	0	0	0	0	0	–	0	–	–
v	–	–	+	+	+	0	–	–	+	0	+	0	0	0	0	0	0	+	0	–	–
vʲ	–	–	+	+	+	0	–	–	+	0	+	0	0	0	0	0	0	–	0	–	–
k	–	+	–	–	–	–	–	–	–	+	0	0	0	0	0	+	–	+	0	–	–
kʲ	–	+	–	–	–	–	–	–	–	+	0	0	0	0	0	+	–	–	0	–	–
g	–	+	–	–	–	–	–	–	+	+	0	0	0	0	0	+	–	+	0	–	–
gʲ	–	+	–	–	–	–	–	–	+	+	0	0	0	0	0	+	–	–	0	–	–
x	–	+	–	–	+	+	–	–	–	+	0	0	0	0	0	+	–	+	0	–	–
xʲ	–	+	–	–	+	+	–	–	–	+	0	0	0	0	0	+	–	–	0	–	–
t	–	+	–	–	–	–	–	–	–	0	0	+	–	+	–	0	0	+	0	–	–
tʲ	–	+	–	–	–	–	–	–	–	0	0	+	–	+	–	0	0	–	0	–	–
d	–	+	–	–	–	–	–	–	+	0	0	+	–	+	–	0	0	+	0	–	–
dʲ	–	+	–	–	–	–	–	–	+	0	0	+	–	+	–	0	0	–	0	–	–
ts	–	+	–	–	–	+	–	–	–	0	0	+	+	+	–	0	0	+	0	–	–
s	–	+	–	–	+	+	–	–	–	0	0	+	+	+	–	0	0	+	0	–	–
sʲ	–	+	–	–	+	+	–	–	–	0	0	+	+	+	–	0	0	–	0	–	–
z	–	+	–	–	+	+	–	–	+	0	0	+	+	+	–	0	0	+	0	–	–
zʲ	–	+	–	–	+	+	–	–	+	0	0	+	+	+	–	0	0	–	0	–	–
ʂ	–	+	–	–	+	+	–	–	–	0	0	+	+	–	+	0	0	+	0	–	–
ʐ	–	+	–	–	+	+	–	–	+	0	0	+	+	–	+	0	0	+	0	–	–
tɕ	–	+	–	–	–	+	–	–	–	0	0	+	+	–	–	0	0	–	0	–	–
ɕ	–	+	–	–	+	+	–	–	–	0	0	+	+	–	–	0	0	–	0	–	–
ʑ	–	+	–	–	+	+	–	–	+	0	0	+	+	–	–	0	0	–	0	–	–
n	–	+	–	+	–	0	–	+	+	0	0	+	–	+	–	0	0	+	0	–	–
nʲ	–	+	–	+	–	0	–	+	+	0	0	+	–	+	–	0	0	–	0	–	–
r	–	+	+	+	+	0	–	–	+	0	0	+	–	+	–	0	0	+	0	–	–
rʲ	–	+	+	+	+	0	–	–	+	0	0	+	–	+	–	0	0	–	0	–	–
l	–	+	+	+	+	0	+	–	+	0	0	+	–	+	–	0	0	+	0	–	–
lʲ	–	+	+	+	+	0	+	–	+	0	0	+	–	+	–	0	0	–	0	–	–
j	–	–	+	+	+	0	0	–	+	0	0	+	–	–	–	+	0	–	0	–	–
i	+	–	+	+	+	0	0	–	+	0	0	0	0	0	0	+	–	–	–	–	–
e	+	–	+	+	+	0	0	–	+	0	0	0	0	0	0	–	–	–	–	–	–
a	+	–	+	+	+	0	0	–	+	0	0	0	0	0	0	–	+	+	–	–	–
o	+	–	+	+	+	0	0	–	+	0	0	0	0	0	0	–	–	+	–	–	–
u	+	–	+	+	+	0	0	–	+	0	0	0	0	0	0	+	–	+	–	–	–
í	+	–	+	+	+	0	0	–	+	0	0	0	0	0	0	+	–	–	+	–	–
é	+	–	+	+	+	0	0	–	+	0	0	0	0	0	0	–	–	–	+	–	–
á	+	–	+	+	+	0	0	–	+	0	0	0	0	0	0	–	+	+	+	–	–
ó	+	–	+	+	+	0	0	–	+	0	0	0	0	0	0	–	–	+	+	–	–
ú	+	–	+	+	+	0	0	–	+	0	0	0	0	0	0	+	–	+	+	–	–
oF	+	–	+	+	+	0	0	–	+	0	0	0	0	0	0	–	–	+	–	+	–
eF	+	–	+	+	+	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	0	0	+

Table 16: Phonological feature matrix employed to calculate phonotactic scores in a Durvasula (2026)-like model.

Quantifying the cross-linguistic effects of syncretism on agreement attraction

Utku Turk
University of Maryland, College Park
utkuturk@umd.edu

Eva Neu
University of Massachusetts, Amherst
eneu@umass.edu

Abstract

Agreement attraction errors, in which a verb erroneously agrees with an intervening noun rather than its grammatical head, are amplified by morphological syncretism in some languages (English, German, Russian) but not others (Turkish, Armenian), a cross-linguistic pattern without a principled account. We use surprisal and attention entropy from large language models as processing proxies to investigate this variation across four languages. LLM-derived measures replicate behavioral findings in English and German (syncretism modulates attraction), align with Turkish null results (no modulation), and partially capture Russian patterns. We discuss further directions for better understanding why syncretism affects agreement attraction differently across languages.

1 Introduction

1.1 Agreement attraction and cue-based retrieval

Consider the following sentences in (1):

(1) a. * The key to the cabinet are rusty.
b. * The key to the cabinets are rusty.

Both sentences are ungrammatical because the verb incorrectly bears plural rather than singular marking. Nonetheless, psycholinguistic research has shown that speakers are more likely to produce, and consider grammatical, sentences like (1b), where the initial preamble includes a plural noun (Wagers et al., 2009). These systematic errors are known as agreement attraction: instead of agreeing with the head noun—here, *key*—, the verb erroneously agrees with an intervening attractor, *cabinets*.

One theory that explains agreement attraction effects is the cue-based retrieval account (Wagers et al., 2009; Dillon et al., 2013). Following the ACT-R model of sentence processing, this model assumes that words are stored in a content-addressable memory as a chunk that can be later retrieved using specific cues (Lewis and Vasishth, 2005). For instance, in (1b), the noun *key* might be stored as the chunk [SG, NOM, TP], encoding number, overt case marking and the dominating syntactic position. Similarly, the word *cabinets* might be stored as [PL, NOM, PP].

According to the account put forward by Wagers et al. (2009), readers form a prediction about the form in which the verb should surface based on the chunks they have processed first. If the noun that occupies the syntactic position of an agreement controller [TP] is singular, they predict that the verb surfaces with singular agreement. In cases where participants indeed see a verb with singular marking such as *is*, no disruption in reading or judgment is seen.[1]

On the other hand, if participants encounter a plural verb such as *are* that does not match with the predicted number, a repair process is initiated. In this process, participants use the relevant cues provided by the verb to search for an agreement controller. The verb *are*, which must agree with a plural noun phrase in subject position bearing nominative case, would trigger the search for a [PL, NOM, TP] chunk. In (1b), no noun phrase matches this chunk perfectly. Accordingly, most of the time participants do not find these sentences grammatical. However, occasionally but systematically, participants erroneously retrieve *cabinets*, encoded as [PL, NOM, PP], as the agreement controller due to a partial match with the cues provided by *are*. In contrast, participants are much less likely to erroneously retrieve the word *cabinet* in (1a), since the

[1] Other models of agreement attraction exist and make different predictions on processing grammatical sentences, most notably Marking and Morphing (Eberhard et al., 2005; Hammerly et al., 2019). We do not focus on these here; see Yadav et al. (2023) for an overview and computational comparison. These models are better suited to account for semantic integration (Solomon and Pearlmutter, 2004), as well as collectivity and distributivity effects (Humphreys and Bock, 2005) that are also known to modulate both agreement and agreement attraction.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 160–170
July 3-4, 2026 ©2026 Association for Computational Linguistics

features shared between the cues of *are* [PL, NOM, TP] and the chunk of *cabinet* [SG, NOM, PP] are even fewer.

1.2 The effect of syncretism

Two grammatical forms are said to be syncretic if they are realized with the same overt morphology despite bearing different syntactic and semantic features. E.g., most noun phrases in English—such as *the dog*—are syncretic between nominative and accusative case marking. An exception are some pronouns which differ in their nominative and accusative forms, as with the first person pronoun *I* (nominative) vs. *me* (accusative).

Under the cue-based retrieval model, it has been argued that in the case of syncretism, a noun phrase is encoded with all possible features that could be realized by its morphology. E.g., *the dog* would be encoded as bearing both NOM and ACC features despite the fact that it only ever bears one of them in the context of a specific sentence. In contrast, *I* would be encoded with only NOM features, and *me* with only ACC features. Given the broader assumptions about cue-based retrieval outlined above, this predicts that an attractor is more likely to interfere with the processing of subject-verb agreement if it is syncretic with the target.

By way of example, such an effect of syncretism on agreement attraction rates can be observed in a study by Hartsuiker et al. (2003) on German. The German plural determiner surfaces as *die* in both the nominative and the accusative case, but as a distinct form, *den*, in the dative. Using the paradigm in (2-3), Hartsuiker et al. showed that participants made more agreement errors while completing sentences with syncretic determiners (*die*).

(2) Die Stellungnahme zu den$_{\text{DAT}}$ Demonstrationen
The position on the demonstrations

(3) Die Stellungnahme gegen die$_{\text{ACC}}$ Demonstrationen
The position against the demonstrations

Under the cue-based retrieval model, this is as expected: since the determiner *die* in (3) is encoded as both ACC and NOM, *die Demonstrationen* is more likely to be misretrieved than the attractor in (2), *den Demonstrationen*, which is only encoded as DAT.

1.3 Cross-linguistic differences

While an effect of syncretism on agreement attraction can be accounted for from the perspective of cue-based retrieval, the cross-linguistic picture is more complex. Syncretism has been shown to increase agreement attraction in English (Nicol et al., 2016), Slovak (Badecker and Kuminiak, 2007) and Czech (Lacina et al., 2025), besides German as seen above. However, in Turkish (Türk and Logačev, 2024) and Armenian (Avetisyan et al., 2020), syncretism does not appear to affect error rates or reading times, respectively. Moreover, French shows a reverse pattern, with syncretic attractors producing *fewer* errors (Franck et al., 2006). Lastly, in addition to normal syncretism effects like English, Russian also shows interference from pseudoplural attractors, which are singular, but syncretic with a plural form (Slioussar, 2018).

To the best of our knowledge, this cross-linguistic variation in the effect of syncretism has yet to receive a principled account. One tentative explanation, suggested by Dillon and Keshev (2025), is that languages might differ as to how useful morphological cues are for resolving agreement. If speakers rely heavily on morphological marking to track agreement dependencies, error rates will highly depend on whether or not target and attractor are syncretic. On the other hand, if speakers rather rely on other factors such as word order or semantics to establish a relation between the verb and its subject, syncretism should have little effect. While plausible, this explanation still lacks empirical confirmation.

Here, we use LLMs as a cross-linguistically uniform processing proxy to quantify syncretism effects on agreement attraction across four languages (English, German, Russian, Turkish), extracting surprisal at the verb and attention entropy (i.e., how diffusely the model attends to candidate nouns at the verb) over potential agreement controllers. We find that LLM-derived measures replicate behavioral data in English, German, and Turkish, and partially in Russian. The one case where our LLM results are clearly at odds with the behavioral measures—Russian pseudoplurals (Slioussar, 2018)—suggests that humans are more easily led astray by superficial morphological similarity, whereas the LLMs draw more heavily on syntactic information from the broader context of the sentence which clearly disambiguates the morphological marking (see also Bazhukov et al., 2024, for similar partial results).

2 Methods

A core finding in computational psycholinguistics is that the difficulty of processing a word, as measured by reading times, is proportional to its surprisal—the negative log probability of that word given its context (Hale, 2001; Levy, 2008). This relationship has been robustly demonstrated across reading paradigms: surprisal estimates from language models correlate linearly with self-paced reading times and eye-tracking measures such as gaze duration and total reading time (Smith and Levy, 2013; Goodkind and Bicknell, 2018). Critically, surprisal from neural language models like GPT-2 has been shown to predict human reading behavior as well as or better than traditional n-gram models (Wilcox et al., 2020; Merkx and Frank, 2021).

This general correlation between reading times and surprisal values also applies to agreement attraction data. Ryu and Lewis (2021) have argued that LLMs provide a correlate for the dynamics of memory retrieval according to the cue-based retrieval account, previously discussed in Section 1.1 (but see also Arehalli and Linzen, 2020). They showed that agreement attraction effects, as measured experimentally in reading times, can be replicated in LLMs using surprisal values, which quantify how likely a word is to appear in a context (Levy, 2008). Concretely, in an ungrammatical sentence like (1b), **The key to the cabinets are rusty*, lower surprisal at the verb corresponds to a stronger effect of agreement attraction since it indicates the possibility that the model is, like humans, led astray by the plural attractor and *illusioned* into deeming the sentence grammatical. In terms of cue-based retrieval, decreased surprisal corresponds to the facilitation due to a partial match between the agreement bearer *are* and the attractor *cabinets*. This partial match would also predict an increased retrieval of the *cabinets* as the agreement controller compared to the cases with singular attractor (1a).

In addition to surprisal, Ryu and Lewis argue that behavioral agreement attraction data also correlate with attention values in an LLM which quantify the importance of a specific word in the context (Clark et al., 2019). How much attention the model attributes, at the point of the verb, to a given noun indicates how likely it considers this noun to be the subject that is related to the verb by an agreement dependency. Accordingly, the more attention the model attributes to the intervening attractor relative to the true target of agreement, the stronger the agreement attraction effect.

Ryu and Lewis's work has focused on linking attention and surprisal in LLMs specifically to reading time data on agreement attraction. However, in experimental studies, including some that we draw on in the present work, agreement attraction is often quantified using offline behavioral measures like acceptability judgments rather than reading times. Crucially, the memory-retrieval assumptions underlying Ryu and Lewis's framework are not paradigm-specific: cue-based retrieval makes predictions about processing difficulty in general, and retrieval interference effects are routinely observed across both online and offline measures in psycholinguistics (Wagers et al., 2009; Dillon et al., 2013). Some recent work further supports the offline–LLM link directly (Lee et al., 2024; Timkey and Linzen, 2023), and we treat our measures as general-purpose proxies for retrieval competition rather than as direct models of any specific experimental paradigm. We note that some of the behavioral datasets we model come from production paradigms, while LLM-based measures are mainly used on the comprehension side (cf. Harmon and Kapatsinski, 2021). We hope to complement the current findings with self-paced reading and acceptability judgment studies to complete the cross-paradigm picture.

In the following, we describe our methodology for extracting surprisal and attention values more in detail.

Surprisal We estimated surprisal for the agreement-bearing words (auxiliary or main verbs) using GPT-2-style autoregressive language models. Because surprisal is defined as the negative log probability of a word given its preceding context, autoregressive models that generate text left-to-right provide the most theoretically appropriate estimate.

Attention We extracted attention weights from BERT-style bidirectional models, measuring the soft attention from the verb position to each noun that could either correctly (grammatical head) or erroneously (attractor) control agreement. Following methodological insights from probing studies (Voita et al., 2019; Clark et al., 2019), we identified syntactically relevant attention heads rather than aggregating across all heads and layers.

Our approach adapts the entropy metric from Ryu and Lewis (2021), who aggregate attention

entropy across syntactically selected heads spanning all layers. Oh and Schuler (2022) instead restrict computation to the final layer, but Ryu and Lewis note that none of Oh and Schuler (2022) selected syntactic heads fall in the final layer, suggesting that intermediate layers are more informative for grammatical dependencies. We follow this insight by using probing to select the most informative layer rather than defaulting to the final one, while keeping the layer-level aggregation of Oh and Schuler (2022). Concretely, to identify which layer best tracks subject-verb dependencies, we parsed 1M-sentence corpora from the Leipzig Corpora Collection for each language using Universal Dependencies (De Marneffe et al., 2021) and selected the layer where the subject most reliably appeared among the verb's five most-attended tokens: layer 6 for English (63.1% accuracy), layer 9 for German (48.2%), layer 8 for Russian (69.3%), and layer 8 for Turkish (53.0%). Attention entropy was then computed over the mean attention distribution across all heads in that layer on the experimental stimuli.

In what follows, we focus on ungrammatical sentences and ask whether adding a plural attractor changes the measure relative to a singular attractor. We compare sentence pairs that only differ in the number marking of the attractor. Our measure of interest is $\Delta = \mu_{\text{Plural}} - \mu_{\text{Singular}}$, computed separately for syncretic and non-syncretic conditions. A more negative Δ for surprisal means the plural attractor made the ungrammatical verb less surprising — i.e., stronger agreement attraction. A more positive Δ for attention entropy means attention was more dispersed, reflecting greater retrieval competition.

3 Materials and results

For inference, we fit per-language Bayesian mixed-effects models separately for surprisal and attention entropy. Each model includes all main effects and interactions among Syncretism, Grammaticality, and Attractor Number as fixed effects. The random-effects structure accounts for by-item variability. Items were allowed to vary in their intercepts as well as their slopes for Grammaticality, Attractor Number, and their interaction, with these random effects treated as uncorrelated. Fixed effects use treatment coding with reference levels Syncretic (Syncretism), Grammatical (Grammaticality), and Singular (Attractor). Under this coding, the two-way term $\beta_{G\times A}$ indexes the Grammaticality×Attractor interaction in the Syncretic baseline, and the three-way term $\beta_{S\times G\times A}$ indexes how that interaction changes in the Non-syncretic condition (thus, the Non-syncretic interaction is $\beta_{G\times A} + \beta_{S\times G\times A}$). In the main text, we report posterior probabilities $P(\text{effect}>0)$ for (i) the two-way Grammaticality×Attractor interaction (attraction effect) and (ii) the three-way Syncretism×Grammaticality×Attractor interaction (syncretism modulation of attraction). Full model details and priors are reported in the Appendix.

3.1 Experiment 1: German

Materials For German, we modeled the data by Hartsuiker et al. (2003) summarized in Section 1.2. Recall that this experiment found that participants made more agreement errors in production when completing sentences with accusative attractors that are syncretic with the nominative (marked with the determiner *die*) compared to dative, non-syncretic attractors (marked with *den*). We created grammatical (singular) and erroneous (plural) continuations for each experimental condition and modeled these data using `bert-base-german-cased` and `dbmdz/german-gpt2`. We expected to find overall lower surprisal and higher attention to the plural attractor in the ungrammatical completions, compared to singular attractor conditions. More importantly, we further predicted that this difference would be amplified in the syncretic condition (2) compared to the non-syncretic condition (3).

Surprisal (Fig.1) Ungrammatical sentences showed higher surprisal compared to grammatical sentences as expected, and singular attractor conditions in ungrammatical sentences showed higher surprisal compared to plural attractors. The important comparison is whether the difference between plural and singular attractors is larger in syncretic than in non-syncretic items. The differences are very similar: non-syncretic $\Delta = -0.66$ (SE=0.35) and syncretic $\Delta = -0.60$ (SE=0.32), so the syncretism modulation in descriptive terms is small ($\Delta_{\text{syn}} - \Delta_{\text{non}} = 0.06$). However, there seems to be an overall syncretism effects, such that non-syncretic nouns induced more surprisal overall. Posterior probabilities still show a clear attraction interaction in the syncretic baseline ($P(\beta_{G\times A}>0)<0.01$), while the three-way term is

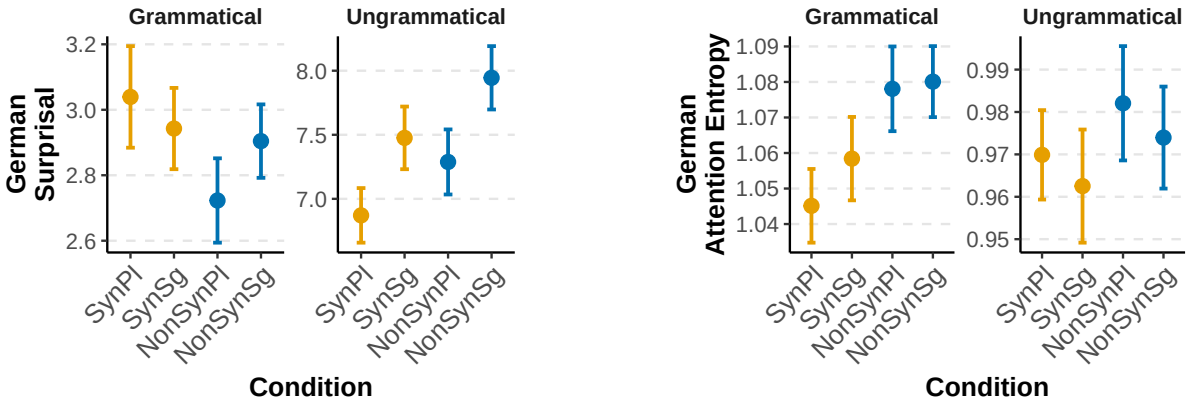


Figure 1: German model-derived surprisal and attention entropy measures (means with SE bars).

positive but uncertain ($P(\beta_{S\times G\times A}>0)$=0.822).

Attention Entropy (Fig.1) Entropy shows the same pattern but much weaker: non-syncretic Δ=0.008 (SE=0.018) and syncretic Δ=0.007 (SE=0.017), with near-zero descriptive modulation (-0.001). Posterior probabilities suggest a positive attraction interaction in the syncretic baseline ($P(\beta_{G\times A}>0)$=0.969), but no reliable three-way interaction ($P(\beta_{S\times G\times A}>0)$=0.250).

Discussion Behaviorally, German is expected to show stronger attraction in syncretic (*die*) than non-syncretic (*den*) conditions. In our current model outputs, both attention and surprisal effects align with attraction predictions, but these values underpredict the syncretic-vs.-non-syncretic contrast.

3.2 Experiment 2: English

Materials For English, we modeled data from Nicol et al. (2016) and Wagers et al. (2009) using pretrained GPT2-small (Radford et al., 2019) and BERT (Devlin et al., 2019). Wagers et al. (2009) found that participants were more likely to produce agreement attraction errors with plural attractors (e.g., *cabinets*) compared to singular attractors (e.g., *cabinet*). However, as noted above, nearly all English nouns are syncretic between nominative and accusative; thus, Wagers et al.'s experiment was limited to attractors that were syncretic between nominative and accusative case marking. Nicol et al. (2016) further investigated whether the syncretism of the attractor is crucial for agreement attraction effects. They compared accusative plural attractors whose case is syncretic with the nominative (e.g., *gardens*) to genitive plural attractors which do not show this syncretism (e.g., *elves'*) and found significantly higher interference rates with syncretic attractors. In their experiment, participants produced fewer agreement attraction errors with preambles like (4c), which contain only a non-syncretic plural attractor (*elves'*), than with (4b) or (4d), which contain a syncretic plural attractor (*gardens*). Moreover, there was no significant difference between (4b) and (4d), indicating that the non-syncretic attractor did not increase interference rates when a syncretic attractor was present.

(4) a. The statue in the elf's garden ...
b. The statue in the elf's gardens ...
c. The statue in the elves' garden ...
d. The statue in the elves' gardens ...

Surprisal (Fig.2) For English, we focus on how much adding plural marking on a possible attractor (either *elf* or *garden*) changes processing in each morphology type: syncretic (*elves' garden* $\rightarrow$ *elves' gardens*) vs. non-syncretic (*elf's gardens* $\rightarrow$ *elves' gardens*). Surprisal was sensitive to grammaticality as expected. Moreover, in ungrammatical items, adding plural marking lowers surprisal much more in syncretic items ($\Delta=-2.009$, SE=0.286) than in non-syncretic items ($\Delta=-0.541$, SE=0.235). This appears as a strongly negative two-way interaction ($P(\beta_{G\times A}>0)<0.01$) plus a strongly positive three-way interaction ($P(\beta_{S\times G\times A}>0)>0.99$); i.e., the attraction pattern is clearly attenuated in non-syncretic items.

Attention Entropy (Fig.2) Entropy shows the same descriptive direction: in ungrammatical items, plural increases entropy in syncretic items (Δ=0.095, SE=0.042), but is near-zero in non-syncretic items (Δ=0.002, SE=0.049). At model level, the syncretic-baseline two-way term is strongly positive ($P(\beta_{G\times A}>0)$=0.978), while the three-way term is mostly negative

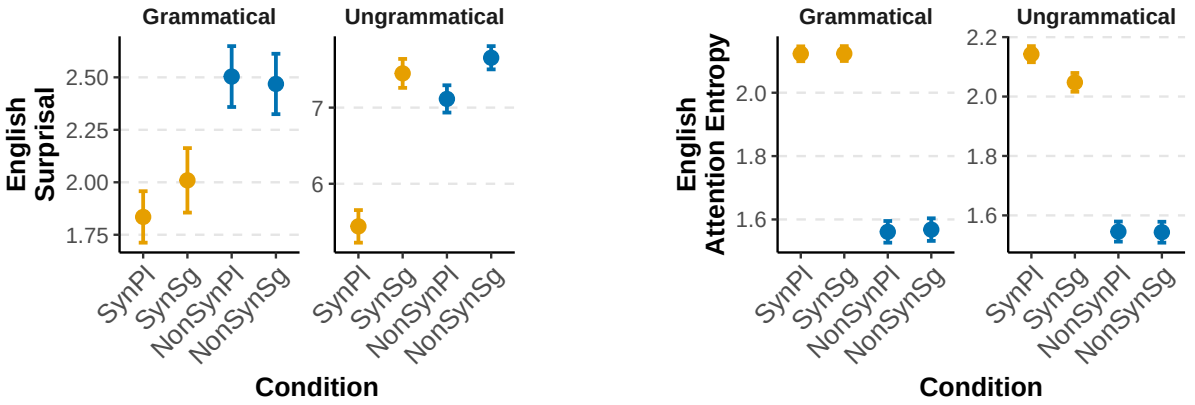


Figure 2: English model-derived surprisal and attention entropy measures (means with SE bars).

($P(\beta_{S\times G\times A}>0)$=0.090), again consistent with attenuation in non-syncretic items.

Discussion Research on English has found higher agreement attraction rates with syncretic (accusative plural) than with non-syncretic attractors (genitive plural). Our results from large language models correlate with these findings. Both surprisal and attention-related values showed a general effect of attraction results, but this effect was also modulated by the presence of syncretic nouns. The important difference in surprisal values increased with non-syncretic nouns, rendering non-syncretic plural meaningless in terms of facilitation effects. Similarly, the non-syncretic nouns also decreased attention entropy, meaning that the directed attention to the head was more concentrated, aligning with increased accuracy.

3.3 Experiment 3: Russian

Materials For Russian, we modeled data from Slioussar (2018) using `deepvk/bert-base-uncased` and `ai-forever/rugpt3small_based_on_gpt2`. In some respect, Russian aligns with German and English in that speakers are more likely to wrongly consider a sentence with a singular head, a plural attractor and a plural verb grammatical if the case marking on the attractor is ambiguous. Slioussar (2018) showed this by comparing accusative attractors such as *polja* ('field.ACC.PL'), which are syncretic with their nominative form (5), to genitive attractors such as *večerinok* ('party.GEN.PL'), which are not ambiguous with nominative marking (6). They found that the syncretic attractors resulted in higher error rates.

(5) Trassa čerez polja/pole byli ...
highway.NOM.SG across field.ACC.PL/SG were ...
'The highway across the fields/field were ...'

(6) Komnata dlja večerinok/večerinki byli ...
room.NOM.SG for party.GEN.PL/SG were
'The room for parties/party were ...'

However, Russian also exhibits another syncretism that affects agreement attraction rates, which differs from the patterns described so far. In (6), the genitive singular form of 'party', *večerinki*, is syncretic with the nominative plural form, whereas the genitive plural form *večerinok* does not exhibit a syncretism with the nominative. This pattern is widespread in Russian nominal paradigms, so any resulting effects cannot be attributed to rarity. Slioussar found that with genitive attractors as in (6), participants produced *more* agreement attraction errors with singular than with plural attractors, contrary to the usual pattern. The effect was robust in both production and comprehension, and visible in reading times. What this suggests is that attractors can be misretrieved upon encountering a plural verb not only if they truly bear plural marking, but also if they are singular but syncretic with a plural form.

Following Slioussar's results, we expect to find a slightly different picture compared to English and German. We expect to see a two-way interaction that signals attraction effects in syncretic cases, and a three-way interaction showing a clear difference between syncretic and non-syncretic cases. However, we also expected to see a reversed attraction effect only in non-syncretic (genitive) cases, corresponding to higher interference rates from genitive singular than genitive plural attractors. To this end, we fitted an additional model only for the non-syncretic cases to see a non-interaction or reversed

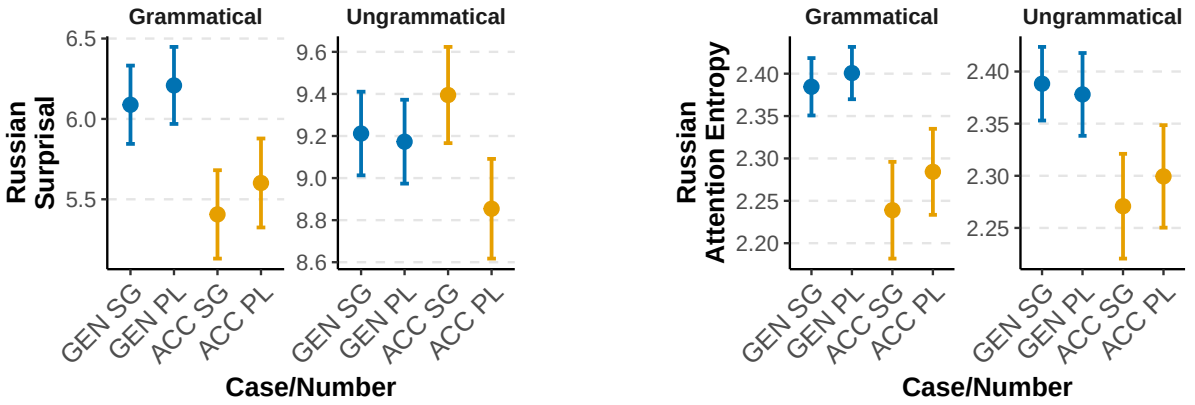


Figure 3: Russian model-derived surprisal and attention entropy measures (means with SE bars).

interaction.

Surprisal (Fig.3) In Russian syncretic (ACC) ungrammatical items, adding a plural attractor lowers surprisal relative to singular ($\Delta_{\text{Pl}-\text{Sg}} = -0.541$, SE=0.329). For this syncretic baseline case, the main model gives a strongly negative two-way interaction ($\beta_{G\times A} = -0.730$, $P(\beta_{G\times A}>0)<0.01$). In non-syncretic (GEN) ungrammatical items, the descriptive contrast is much smaller and close to reversal: $\Delta_{\text{Pl}-\text{Sg}} = -0.039$ (SE=0.282). The main-model three-way interaction is strongly positive ($\beta_{S\times G\times A}=0.568$, $P(\beta_{S\times G\times A}>0)>0.99$), and the non-syncretic-only model gives a negative two-way interaction ($\beta_{G\times A} = -0.159$, $P(\beta_{G\times A}>0)<0.01$), consistent with the reversal pattern, though the magnitude is very small compared to Slioussar's (2018) results.

Attention entropy (Fig.3) In Russian syncretic (ACC) ungrammatical items, adding a plural attractor slightly increases entropy relative to singular ($\Delta_{\text{Pl}-\text{Sg}}=0.029$, SE=0.070). For this syncretic baseline case, the main-model two-way term is weak and not clearly directional ($\beta_{G\times A} = -0.016$, $P(\beta_{G\times A}>0)=0.183$). In non-syncretic (GEN) ungrammatical items, the descriptive effect is reversed and near zero ($\Delta_{\text{Pl}-\text{Sg}} = -0.010$, SE=0.053; equivalently $\Delta_{\text{Sg}-\text{Pl}} = +0.010$). The main-model three-way interaction is also weak ($\beta_{S\times G\times A} = -0.010$, $P(\beta_{S\times G\times A}>0)=0.352$), and the non-syncretic-only two-way term is similarly weak ($\beta_{G\times A} = -0.025$, $P(\beta_{G\times A}>0)=0.138$).

Discussion Our LLM based values show clear attraction effect within syncretic items. In non-syncretic cases, we find that it is heavily attenuated, similar to the English findings. While this overall effect is expected, Russian behavioral data also show a reversed effect within the GEN conditions in which the singular attractor leads to higher interference rates. However, this reversal is not fully supported, with much weaker non-syncretic (GEN) contrasts and weak non-syncretic-only reversed-attraction effects. Entropy remains less decisive overall.

3.4 Experiment 4: Turkish

Materials For Turkish, we modeled data from Türk and Logačev (2024), who compared agreement attraction rates between sentences such as (7) and (8), building on work by Lago et al. (2019). We used `dbmdz/bert-base-turkish-128k-cased` and `redrussianarmy/gpt2-turkish-cased`. The head noun (here, *eğitmen-i* 'instructor' or *hoca-sı* 'teacher') bears a possessor morpheme and is preceded by a genitive-marked possessee (here, *teknisyen-(ler)-in* 'of the technician(s)').

(7) Teknisyen-(ler)-in eğitmen-i … koştu-(lar).
tech-(PL)-GEN instructor-POSS ran-(PL)
'The instructor of the technician(s) … $\text{ran}_{\text{SG/PL}}$.'

(8) Teknisyen-(ler)-in hoca-sı … koştu-(lar).
tech-(PL)-GEN teacher-POSS ran-(PL)
'The teacher of the technician(s) … $\text{ran}_{\text{SG/PL}}$.'

Since genitive case also surfaces on the subjects of embedded clauses in Turkish, speakers might misinterpret the possessee as an embedded subject and, if it is plural, misretrieve it when encountering a plural verb. Crucially, after a consonant as in (7), the possessor morpheme on the head is realized as *-I*, which is syncretic with accusative case. Vowel-final nouns as in (8), on the other hand, take the possessor morpheme *-sI*, which is not syncretic but can only surface on nominative subjects.[2] Türk

[2]Note that in the Turkish materials, the attractor precedes

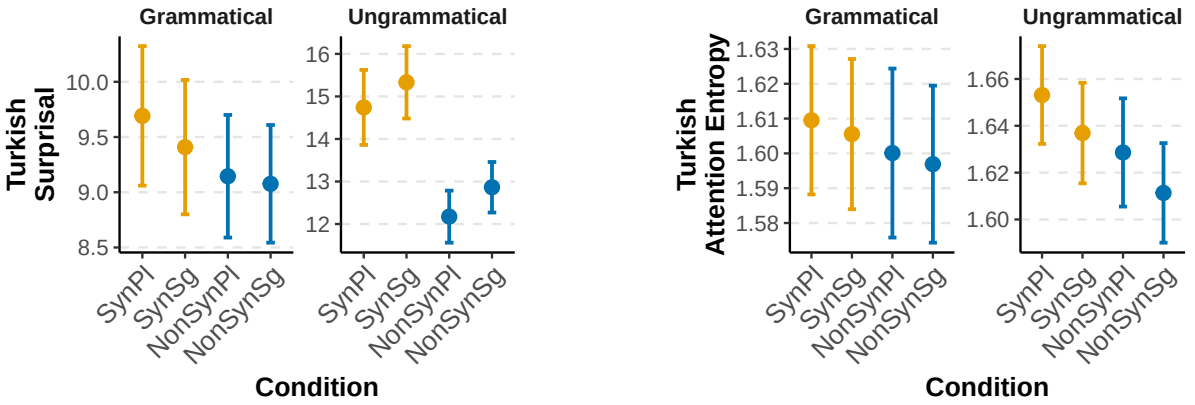


Figure 4: Turkish model-derived surprisal and attention entropy measures (means with SE bars).

and Logačev (2024) expected that the unambiguous nominative case marking on the head as in (8) (*hoca-sı*) would diminish interference rates from the genitive attractor *teknisyen-ler-in*, but this prediction was not borne out. Unlike in the languages seen so far, syncretism does not appear to affect agreement attraction rates in Turkish.

Following their results, we expect that plural attractors overall decrease surprisal and increase attention entropy in ungrammatical sentences. However, this effect should not be modulated by syncretism.

Surprisal (Fig.4) For Turkish, the key test is whether plural-attractor effects differ by syncretism. They do not: adding a plural attractor lowers surprisal in both conditions, with the effects being of very similar size (non-syncretic $\Delta = -0.691$, SE=0.853; syncretic $\Delta = -0.590$, SE=1.230). Bayesian estimates show weak directional evidence for the syncretic-baseline two-way term ($P(\beta_{G\times A}>0)$=0.194), and more importantly, no evidence for a three-way interaction ($P(\beta_{S\times G\times A}>0)$=0.332).

Attention entropy (Fig.4) Entropy shows the same qualitative conclusion. The plural-attractor increase is almost identical in the two conditions (non-syncretic Δ=0.017, SE=0.031; syncretic Δ=0.016, SE=0.030). However, posterior probabilities show no directional evidence for both interaction terms ($P(\beta_{G\times A}>0)$=0.782, $P(\beta_{S\times G\times A}>0)$=0.550).

Discussion While our surprisal results align with the behavioral Turkish data—namely, less surprisal in agreement attraction cases—the attention values are inconclusive. As for the syncretism, we show that it does not modulate attraction related facilitation within surprisal, and plural-attractor effects are comparable in syncretic and non-syncretic conditions. Given that there is a general sensitivity in LLM measures to attraction effects, the lack of syncretism modulation is consistent with the behavioral data.

4 General discussion

This study aimed to establish a baseline for future work on syncretism and morphological encoding in memory retrieval by identifying correlates between human behavioral data and LLM-based measures, following recent methodological advances (Wilcox et al., 2020; Ryu and Lewis, 2021). Across three of the four languages examined, model-derived measures matched the effects attested in the behavioral literature. Surprisal provided a clearer and more consistent signal, while attention entropy was less decisive: the expected patterns were present but weaker and less certain. This divergence is not unexpected, given that attention-based measures are more indirect and less established than surprisal as proxies for retrieval competition, but it points to room for improvement in how subject-tracking attention is extracted and interpreted.

The one clear exception is Russian. Surprisal successfully captured the standard attraction effect in the accusative (syncretic) condition, but failed to

the head noun, whereas in the English, German, and Russian materials, the attractor intervenes between the head and the verb. At least for Russian and German, and possibly Turkish, alternative orders can be constructed (e.g., via topicalization or focus movement) without changing the structure drastically. Sturt and Kwon (2024) found that distractor position modulated the size of attraction effects in English; however, whether similar positional effects hold across languages and whether LLMs mirror human behavior under these rarer configurations remains an open question.

reproduce the reversed pattern in the genitive condition, where behavioral data show that a genitive singular attractor—syncretic with the nominative plural—generates more interference than a genuine genitive plural. One possible explanation is that LLMs are less sensitive than human comprehenders to spurious morphological overlaps between two forms that share no underlying feature ([GEN, SG] and [NOM, PL]) and/or rely more heavily on syntactic information to disambiguate these forms. Resolving this question requires targeted probing of how models represent syncretism with and without feature overlap, which we leave for future work.

More broadly, we hope that in the long term, this study contributes to understanding of *why* syncretism modulates agreement attraction differently across languages. Our results suggest that LLMs are sensitive to the same syncretisms that drive attraction in human experiments, even if through different underlying mechanisms. A promising direction for future work is to ask whether languages differ systematically in how much weight they place on morphological cues when tracking dependencies. Syncretism effects may be strongest in languages where morphological distinctions are few but highly informative (English, German) or where case encodes a dense bundle of features (fusional languages like Russian and Czech). In agglutinative languages like Turkish and Armenian, where morphological information is more transparent and compositional, speakers and models alike may rely less on the presence or absence of a specific syncretic overlap, leading to the null effects observed here.

References

Suhas Arehalli and Tal Linzen. 2020. Neural language models capture some, but not all, agreement attraction effects. In Proceedings of the Annual Meeting of the Cognitive Science Society.

Serine Avetisyan, Sol Lago, and Shravan Vasishth. 2020. Does case marking affect agreement attraction in comprehension? Journal of Memory and Language, 112:104087.

William Badecker and Frantisek Kuminiak. 2007. Morphology, agreement and working memory retrieval in sentence production: Evidence from gender and case in Slovak. Journal of Memory and Language, 56:65–85.

Maxim Bazhukov, Ekaterina Voloshina, Sergey Pletenev, Arseny Anisimov, Oleg Serikov, and Svetlana Toldova. 2024. Of models and men: Probing neural networks for agreement attraction with psycholinguistic data. In Proceedings of the 28th Conference on Computational Natural Language Learning, pages 280–290.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does BERT look at? An analysis of BERT's attention. arXiv Prepr. arXiv:1906,04341.

Marie-Catherine De Marneffe, Christopher D Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. Computational Linguistics, 47:255–308.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.4805.

Brian Dillon and Maayan Keshev. 2025. Syntactic dependency formation in sentence processing: A comparative perspective. In Sjef Barbiers, Norbert Corver, and Maria Polinsky, editors, Cambridge Handbook of Comparative Syntax. Cambridge UP.

Brian Dillon, Alan Mishler, Shayne Sloggett, and Colin Phillips. 2013. Contrasting intrusion profiles for agreement and anaphora: Experimental and modeling evidence. Journal of Memory and Language, 69:85–103.

Kathleen M Eberhard, J Cooper Cutting, and Kathryn Bock. 2005. Making syntax of sense: number agreement in sentence production. Psychological Review, 112:531.

Julie Franck, Glenda Lassi, Ulrich H Frauenfelder, and Luigi Rizzi. 2006. Agreement and movement: A syntactic analysis of attraction. Cognition, 101:173–216.

Adam Goodkind and Klinton Bicknell. 2018. Predictive power of word surprisal for reading times is a linear function of language model quality. In Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics, pages 10–18.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. 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.

Zara Harmon and Vsevolod Kapatsinski. 2021. A theory of repetition and retrieval in language production. Psychological Review, 128:1112.

Robert J Hartsuiker, Herbert J Schriefers, Kathryn Bock, and Gerdien M Kikstra. 2003. Morphophonological influences on the construction of subject–verb agreement. Memory & Cognition, 31:1316–1326.

Karin R Humphreys and Kathryn Bock. 2005. Notional number agreement in English. Psychonomic Bulletin & Review, 12:689–695.

Radim Lacina, Anna Laurinavichyute, and Jan Chromỳ. 2025. Only case-syncretic nouns attract: Czech and Slovak gender agreement. Journal of Memory and Language, 143:104623.

Sol Lago, Martina Gračanin–Yuksek, Duygu Fatma Şafak, Orhan Demir, Bilal Kırkıcı, and Claudia Felser. 2019. Straight from the horse's mouth: Agreement attraction effects with Turkish possessors. Linguistic Approaches to Bilingualism, 9:398–426.

Eun-Kyoung Rosa Lee, Sathvik Nair, and Naomi Feldman. 2024. A psycholinguistic evaluation of language models' sensitivity to argument roles. In Findings of the Association for Computational Linguistics: EMNLP 2024, pages 3262–3274.

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

Richard L. Lewis and Shravan Vasishth. 2005. An activation–based model of sentence processing as skilled memory retrieval. Cognitive Science, 29:375–419.

Danny Merkx and Stefan L. Frank. 2021. Human sentence processing: Recurrence or attention? Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics.

Janet Nicol, Andrew Barss, and Jason E Barker. 2016. Minimal interference from possessor phrases in the production of subject-verb agreement. Frontiers in Psychology, 7:548.

Byung-Doh Oh and William Schuler. 2022. Entropy- and distance-based predictors from GPT-2 attention patterns predict reading times over and above GPT-2 surprisal. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 9324–9334.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and Others. 2019. Language models are unsupervised multitask learners. OpenAI Blog, 1(8):9.

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 Prepr. arXiv:2104,12874.

Natalia Slioussar. 2018. Forms and features: The role of syncretism in number agreement attraction. Journal of Memory and Language, 101:51–63.

Nathaniel J. Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. Cognition, 128:302–319.

Eric S Solomon and Neal J Pearlmutter. 2004. Semantic integration and syntactic planning in language production. Cognitive Psychology, 49:1–46.

Patrick Sturt and Nayoung Kwon. 2024. Agreement attraction in comprehension: do active dependencies and distractor position play a role? Language, Cognition and Neuroscience, 39:279–301.

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

Utku Türk and Pavel Logačev. 2024. Agreement attraction in Turkish: The case of genitive attractors. Language, Cognition and Neuroscience, 39:448–454.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5797–5808.

Matthew W. Wagers, Ellen F. Lau, and Colin Phillips. 2009. Agreement attraction in comprehension: Representations and processes. Journal of Memory and Language, 61:206–237.

Ethan Wilcox, Jon Gauthier, Jennifer Hu, Peng Qian, and Roger Levy. 2020. On the predictive power of neural language models for human real-time comprehension behavior. In Proceedings of the Annual Meeting of the Cognitive Science Society.

Himanshu Yadav, Garrett Smith, Sebastian Reich, and Shravan Vasishth. 2023. Number feature distortion modulates cue-based retrieval in reading. Journal of Memory and Language, 129:104400.

A Model Specification and Full Results

All Bayesian models were fit with the `brms` package in R (Gaussian family). Priors: $\text{Normal}(0, 2)$ on all fixed effects, $\text{Student-}t(3, 0, 2.5)$ on the intercept, $\text{Exponential}(1)$ on random-effect SDs and residual σ. Sampling: 4 chains $\times$ 12,000 iterations (2000 warmup), `adapt_delta`$=0.95$, `max_treedepth`$=12$, seed 1234. The main model formula was:

```
value ~ Syn * Gram * Attr + (1 + Gram *
            Attr || Item)
```

Treatment coding: Syncretic, Grammatical, and Singular as reference levels, so the intercept represents the expected value for grammatical, syncretic, singular-attractor items. All Russian fits were restricted to singular-head items. Code and data for our paper can be found at `www.github.com/utkuturk/scil_TurkNeu2026`.

Table 1: Full posterior summaries for the **Surprisal** models. Terms are shown as rows; languages as column groups. $\hat{\beta}$ is the posterior mean and P is $P(\text{effect}>0)$. S = Syncretism (Non-syncretic vs. Syncretic), G = Grammaticality (Ungrammatical vs. Grammatical), A = Attractor (Plural vs. Singular). **Bold**: $P>0.89$ or $P<0.11$.

	English		German		Russian		Turkish	
Term	$\hat{\beta}$	P	$\hat{\beta}$	P	$\hat{\beta}$	P	$\hat{\beta}$	P
Intercept	2.091	$>$ **0.999**	2.944	$>$ **0.999**	5.411	$>$ **0.999**	9.793	$>$ **0.999**
S	0.351	**0.972**	−0.044	0.339	0.611	**0.953**	−0.795	**0.015**
G	5.358	$>$ **0.999**	4.460	$>$ **0.999**	3.917	$>$ **0.999**	5.481	$>$ **0.999**
A	−0.220	**0.092**	0.085	0.718	0.191	$>$ **0.999**	0.099	0.596
$S\times G$	−0.160	0.244	0.512	$>$ **0.999**	−0.803	**0.009**	−1.739	$<$ **0.001**
$S\times A$	0.270	0.885	−0.272	**0.053**	−0.071	0.207	−0.022	0.482
$G\times A$	−1.731	$<$ **0.001**	−0.690	$<$ **0.001**	−0.730	$<$ **0.001**	−0.488	0.194
$S\times G\times A$	1.131	$>$ **0.999**	0.218	0.822	0.568	$>$ **0.999**	−0.307	0.332

Table 2: Full posterior summaries for the **Attention Entropy** models. Terms are shown as rows; languages as column groups. $\hat{\beta}$ is the posterior mean and P is proportion of posterior samples that are bigger than 0 $P(\text{effect}>0)$. S = Syncretism (Non-syncretic vs. Syncretic), G = Grammaticality (Ungrammatical vs. Grammatical), A = Attractor (Plural vs. Singular). **Bold**: $P>0.89$ or $P<0.11$

	English		German		Russian		Turkish	
Term	$\hat{\beta}$	P	$\hat{\beta}$	P	$\hat{\beta}$	P	$\hat{\beta}$	P
Intercept	2.106	$>$ **0.999**	1.059	$>$ **0.999**	2.237	$>$ **0.999**	1.597	$>$ **0.999**
S	−0.537	$<$ **0.001**	0.022	**0.994**	0.147	**0.987**	−0.006	0.281
G	−0.076	**0.012**	−0.096	$<$ **0.001**	0.032	**0.956**	0.031	$>$ **0.999**
A	0.000	0.513	−0.013	**0.074**	0.045	**0.988**	0.004	0.643
$S\times G$	0.049	0.867	−0.011	0.163	−0.027	0.166	−0.018	0.120
$S\times A$	−0.008	0.411	0.011	0.831	−0.029	0.150	−0.001	0.471
$G\times A$	0.093	**0.978**	0.021	**0.969**	−0.016	0.183	0.012	0.782
$S\times G\times A$	−0.083	**0.090**	−0.010	0.250	−0.010	0.352	0.002	0.550

Table 3: Russian non-syncretic-only models (GEN condition only). No Syncretism factor; intercept is the mean for grammatical, singular-attractor items. $\hat{\beta}$ is the posterior mean and P is proportion of posterior samples that are bigger than 0 $P(\text{effect}>0)$. G = Grammaticality (Ungrammatical vs. Grammatical), A = Attractor (Plural vs. Singular). **Bold**: $P>0.89$ or $P<0.11$

	Surprisal		Entropy	
Term	$\hat{\beta}$	P	$\hat{\beta}$	P
Intercept	6.054	$>$ **0.999**	2.380	$>$ **0.999**
G	3.119	$>$ **0.999**	0.004	0.613
A	0.117	**0.976**	0.016	0.821
$G\times A$	−0.159	$<$ **0.001**	−0.025	0.138

Compiling Search & Change Rules into Subsequential Finite-State Transducers

Malek Azadegan
City University of New York, The Graduate Center
mazadegan@gc.cuny.edu

Abstract

Search & Change (S&C) is a procedural model of phonological rule application that is conceptually clear and linguistically motivated, but whose computational properties have not been fully characterized. This paper provides a formal specification of S&C within the framework of Logical Phonology, presents a linear-time algorithm for rule application with a proof of correctness, and gives a compilation procedure mapping S&C rules to a single transition structure that is subsequential in one scan orientation and reverse-subsequential in the other, situating S&C within a well-understood subclass of regular string-to-string functions with known learnability guarantees and algebraic characterizations, implying that S&C-definable mappings are learnable from positive input/output pairs and amenable to algebraic classification.

1 Introduction

A central goal in computational phonology is to characterize the computational nature of phonological patterns using theorems and proofs (Heinz and Idsardi, 2017). Search & Change (S&C; Dabbous et al., 2021, 2024) is a procedural model of phonological rule application that is well-suited to this treatment. S&C frames phonological processes explicitly: target segments initiate directional scans through a word, looking for triggers, and the target segment is transformed if the first trigger encountered satisfies a licensing condition.

Despite its explicit nature, S&C lacks a precise formal specification: the literature provides no rigorous definition of its data structures, no provably correct algorithm for computing derivations, and no characterization of the functions it expresses. Logical Phonology (LP; Bale et al., 2019; Reiss, 2021; Gorman and Reiss, 2025a), a research program that treats phonological objects as mathematical objects subject to explicit definition, provides the formal language needed to make S&C amenable to formal definition and proofs. Building on this framework, the paper proceeds in three steps. First, the underlying data structures, rule format, and execution semantics of S&C are defined. Second, a linear-time algorithm for rule application is presented with a proof of correctness. Third, a compilation procedure that maps S&C rules to subsequential or reverse-subsequential finite-state transducers is given, placing S&C within a well-understood subclass of regular string-to-string functions with known learnability guarantees and algebraic characterizations. With this roadmap in place, the next section lays out the formal preliminaries.

2 Preliminaries

S&C operates over phonological representations built from three primitives: valued features, segments, and natural classes (Dabbous et al., 2021). The following subsections define these primitives and establish the vocabulary used throughout the paper.

2.1 Segments as Sets of Valued Features

Given a universal, finite feature set $\mathcal{F}$ (Chomsky and Halle, 1968; Reiss and Volenec, 2022), a *valued feature* is a member of $\{+,-\}\times\mathcal{F}$. Following Bale et al. (2014, 2019), to abbreviate valued features, $+$VOICE and $-$BACK are written for $(+,\textsc{Voice})$ and $(-,\textsc{Back})$, respectively.

A *segment* x is a consistent set of valued features such that $\forall f \in \mathcal{F}\ \neg\big(+f \in x \wedge -f \in x\big)$. Allowing underspecification (Gorman and Reiss, 2025a), a segment can be defined as a total function $\sigma : \mathcal{F} \to \{\emptyset,+,-\}$, where $\sigma(f) = \emptyset$ indicates underspecification for feature f. The set-based and function-based representations are isomorphic: a total function σ can be represented as the set

$$\{(v,f) \in \{+,-\}\times\mathcal{F} \mid \sigma(f) = v\}.$$

Under this formulation, set consistency corresponds exactly to functionality: no segment can contain

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 171–179
July 3-4, 2026 ©2026 Association for Computational Linguistics

both $+f$ and $-f$ because σ assigns at most one value to each feature. This lets us use whichever representation is more convenient below: total functions for defining the segment space $\Omega(\mathcal{F})$, and set operations for stating feature-changing transformations. The set of all segments is then defined as

$$\Omega(\mathcal{F}) \;=\; \{\, \sigma \mid \sigma : \mathcal{F} \to \{\emptyset, -, +\} \,\} \qquad (1)$$

where $|\Omega(\mathcal{F})| = 3^{|\mathcal{F}|}$. By convention, underspecified segments are denoted with capital letters (e.g., /U/). For example, the vowel /u/ might be represented as $\{+\textsc{Syllabic}, +\textsc{High}, +\textsc{Back}\}$, while an underspecified /U/ lacking a HIGH value could be $\{+\textsc{Syllabic}, +\textsc{Back}\}$. An inconsistent set such as $\{+\textsc{Back}, -\textsc{Back}\}$ violates the consistency condition and is not a valid segment.

2.2 Natural Classes as Sets of Segments

Natural classes are sets of segments defined by shared valued features (Bale and Reiss, 2015; Bale et al., 2019). For any feature specification $F \subseteq (\{+,-\} \times \mathcal{F})$,[1] the natural class defined by F is

$$\mathcal{N}(F) = \{x \in \Omega(\mathcal{F}) \mid F \subseteq x\}.$$

Conventionally, natural classes are written with square brackets, e.g. $[+\textsc{Syllabic}]$; $\mathcal{N}$-notation is used to make explicit that the class is quantified over $\Omega(\mathcal{F})$, which keeps later algorithms and proofs uniform.[2] Specificity is inclusion on specifications: $F_1 \supseteq F_2$ implies $\mathcal{N}(F_1) \subseteq \mathcal{N}(F_2)$ (Gorman and Reiss, 2025b), from which the definition of the universal natural class $\mathcal{N}(\emptyset) = \Omega(\mathcal{F})$ immediately follows.

2.3 Words as Strings of Segments

Words are strings of segments taken from $\Omega(\mathcal{F})$. Formally, $\Omega(\mathcal{F})^*$ is the free monoid over $\Omega(\mathcal{F})$ under concatenation, whose elements are finite sequences $\langle s_1, \ldots, s_n \rangle$ where each $s_i \in \Omega(\mathcal{F})$, with the empty string ϵ acting as the identity element (Heinz, 2018; Dabbous et al., 2021, 2024). In the examples below, IPA symbols are used as a convenient shorthand for segments as defined above; each symbol abbreviates one feature-valued segment. For example, /kæt/ denotes a word $\langle s_1, s_2, s_3 \rangle$ consisting of the segments written /k/, /æ/, and /t/.

[1] Under the identification of total valuations with their sets of non-empty feature assignments, each $x \in \Omega(\mathcal{F})$ may also be viewed as a subset of $(\{+,-\} \times \mathcal{F})$. Segments can therefore serve as feature specifications and thus as arguments to $\mathcal{N}$.

[2] For example, $[+\textsc{Syllabic}] = \mathcal{N}(\{+\textsc{Syllabic}\})$.

3 Search & Change

S&C is "a model of phonological computation characterized by a small set of parameters" (Dabbous et al., 2021). A S&C rule (Dabbous et al., 2021) is a quintuple $\langle \textsc{Inr}, \textsc{Trm}, \textsc{Dir}, \textsc{Out}, \textsc{Cnd} \rangle$ specifying a target class, a trigger class, a search direction, an output function, and a licensing condition (Dabbous et al., 2024).

3.1 Formalizing Search

The search domain is determined by INR, TRM, and DIR: INR and TRM are natural classes of search initiating and terminating segments, and $\textsc{Dir} \in \{\textsc{Left}, \textsc{Right}\}$ determines the search direction. Fix a word $w = \langle s_1, \ldots, s_n \rangle$ and a set TRM. For each position i, let R_i and L_i denote the sets of terminator indices strictly to the right and left of i, respectively. The search function Θ then returns the index of the nearest terminating segment in direction DIR: $\min R_i$ for rightward search and $\max L_i$ for leftward search, and is undefined when the relevant set is empty. The symbols $\downarrow$ and $\uparrow$ indicate definedness and undefinedness in the usual partial-function sense. For each position i in w:

$$\begin{aligned} R_i &= \{\, j \mid j > i \wedge s_j \in \textsc{Trm} \,\} \\ L_i &= \{\, j \mid j < i \wedge s_j \in \textsc{Trm} \,\} \\ \Theta_R(i) &= \begin{cases} \min R_i & R_i \neq \emptyset \\ \uparrow & \text{otherwise} \end{cases} \\ \Theta_L(i) &= \begin{cases} \max L_i & L_i \neq \emptyset \\ \uparrow & \text{otherwise} \end{cases} \end{aligned} \qquad (2)$$

When defined, $s_{\Theta_R(i)}$ or $s_{\Theta_L(i)}$ denotes the terminating segment for s_i under rightward or leftward search respectively.

3.2 Formalizing Change

While INR, TRM, and DIR determine the search domain, CND and OUT decide whether a target segment is eligible for change and what change applies. CND is a natural class and OUT is a function of type $\Omega(\mathcal{F}) \times \Omega(\mathcal{F}) \to \Omega(\mathcal{F})$.

3.2.1 The Condition

If $\Theta_R(i)$ or $\Theta_L(i)$ is defined, let s_j denote the terminating segment at that index. If $s_j \in \textsc{Cnd}$, the rule is licensed to apply OUT to s_i; otherwise s_i is unchanged.

3.2.2 The Output Function

OUT is defined using three operators over $\Omega(\mathcal{F})$ (Bale et al., 2014; Reiss, 2021; Gorman and Reiss, 2025b):

Subtraction. Removes valued features from segments (Bale et al., 2014):

$$A \setminus B = \{c\text{F} \mid c\text{F} \in A \wedge c\text{F} \notin B\} \tag{3}$$

Subtraction is essentially set difference: only valued features present in A but absent from B survive. It applies vacuously whenever B is empty or shares no elements with A. Consider

1. $\{+\text{F}, +\text{G}, -\text{H}\} \setminus \{+\text{G}\} = \{+\text{F}, -\text{H}\}$
2. $\{+\text{F}, -\text{G}\} \setminus \emptyset = \{+\text{F}, -\text{G}\}$
3. $\{+\text{F}, -\text{G}\} \setminus \{+\text{H}\} = \{+\text{F}, -\text{G}\}$.

Case 1 removes $+$G from A since it appears in B. Cases 2 and 3 are vacuous: in case 2 there is nothing to subtract; in case 3, $+\text{H} \notin A$, so the intersection $A \cap B$ is empty and nothing is removed.

Unification. Adds valued features without creating inconsistency (Bale et al., 2014):

$$A \sqcup B = A \cup \{c\text{F} \mid c\text{F} \in B \wedge -c\text{F} \notin A\} \tag{4}$$

Unlike plain set union, $\sqcup$ skips any feature in B for which A already carries a specification, preventing inconsistency; when A and B disagree on a feature, A takes priority, making $\sqcup$ an instance of priority union (Reiss, 2022) rather than unification in the logical or type-theoretic sense. The term *unification* is retained following Bale et al. (2014). The following cases illustrate the range of outcomes:

1. $\{-\text{F}\} \sqcup \{+\text{G}\} = \{-\text{F}, +\text{G}\}$
2. $\{-\text{F}\} \sqcup \{+\text{F}\} = \{-\text{F}\}$
3. $\{+\text{G}\} \sqcup \{+\text{G}\} = \{+\text{G}\}$

Case 1 succeeds: A has no specification for G, so $+$G is added freely. Cases 2 and 3 are vacuous: in case 2, A already contains the opposite value $-$F, so $+$F is blocked to prevent inconsistency; in case 3, A already contains the same value $+$G, so adding it again changes nothing.

Projection. Restricts a segment to a subset of its valued features (Codd, 1970). Viewing a segment s as a set of valued features $(v, f) \in \{+, -\} \times \mathcal{F}$, projection onto $F \subseteq \mathcal{F}$ retains only those pairs whose feature label f falls in F. Equivalently, it restricts the domain of s to F:

$$\pi_F(s) = s \cap (\{-, +\} \times F) \tag{5}$$

Projection returns the partial specification of a segment restricted to a given set of features, as if the remaining features were unspecified. Using $s = \{+\text{F}, +\text{G}, -\text{H}, +\text{J}\}$:

1. $\pi_{\{\text{G,H}\}}(s) = \{+\text{G}, -\text{H}\}$
2. $\pi_{\{\text{J}\}}(s) = \{+\text{J}\}$
3. $\pi_{\emptyset}(s) = \emptyset$

Case 1 keeps G and H while discarding F and J. Case 2 isolates a single feature. Case 3 projects onto the empty set, yielding the fully underspecified segment $\emptyset$.

Combining Operators. Unification implements feature-filling, while subtraction followed by unification models feature-changing (Bale et al., 2014, 2019). For instance, a devoicing rule uses $(s_i \setminus \{+\text{VOICE}\}) \sqcup \{-\text{VOICE}\}$, while feature copying uses $s_i \sqcup \pi_{\{\text{VOICE}\}}(s_j)$.

Operator Complexity. Since $\mathcal{F}$ is finite, segments can be represented as bitvectors of length $2|\mathcal{F}|$, with two bits per feature $f \in \mathcal{F}$ encoding the presence and polarity of its specification. Under this representation, subtraction, unification, and projection all reduce to bitwise operations and execute in $O(|\mathcal{F}|)$ time, which is constant since $\mathcal{F}$ is fixed.

3.3 The Search & Change Specification

Let $\Theta(i)$ denote $\Theta_R(i)$ or $\Theta_L(i)$ depending on DIR. Say that position i is *licensed* if

$$s_i \in \text{INR} \wedge \Theta(i)\downarrow \wedge\, s_{\Theta(i)} \in \text{CND}.$$

For every position i, the transformed segment is

$$s'_i = \begin{cases} \text{OUT}(s_i, s_{\Theta(i)}) & \text{if } i \text{ is licensed} \\ s_i & \text{otherwise} \end{cases} \tag{6}$$

For an input word $w = \langle s_1, \ldots, s_n \rangle$, the output word is $w' = \langle s'_1, \ldots, s'_n \rangle$, with each s'_i computed simultaneously from w.

Simultaneous Application. All s'_i are computed from the same input word w and applied at once, so intermediate changes cannot feed or bleed within a single rule application, following the Simultaneous Application Principle (Chomsky and Halle, 1968; Kenstowicz and Kisseberth, 1979). This assumption is inherited from the S&C framework itself (Dabbous et al., 2024) and is what makes the specification in Equation 6 well-defined: since all output segments are read from the same input, there is no possibility of conflicting licensed changes. Across rules, derivations proceed by explicit ordering (Chomsky and Halle, 1968; Kenstowicz and Kisseberth, 1979). Under ordered rule application, the output of one compiled transducer serves as the input to the next; thus a derivation corresponds to composition of the compiled transducers in rule order.

4 Worked Examples

Two case studies are presented. First, Georgian clear and dark lateral alternation (Kenstowicz and Kisseberth, 1979) as a local example; and second, Votic vowel harmony (Leduc et al., 2020) as a long-distance example.

4.1 Georgian Clear and Dark Laterals (Local)

Kenstowicz and Kisseberth (1979) observe that lateral "lightness" in Georgian is fully predictable: underlying dark laterals surface as clear before −Back vowels. Setting $\textsc{Trm} = \mathcal{N}(\emptyset)$ models adjacency, since the first segment found is always the immediately adjacent one. The S&C rule

$$R_1 = \langle \textsc{Inr}, \textsc{Trm}, \textsc{Dir}, \textsc{Out}, \textsc{Cnd} \rangle$$
$$\textsc{Inr} = \mathcal{N}(\{+\textsc{Lateral}, +\textsc{Dorsal}\})$$
$$\textsc{Trm} = \mathcal{N}(\emptyset)$$
$$\textsc{Dir} = \textsc{Right}$$
$$\textsc{Out} = \lambda s_i.\, \lambda s_j.\, (s_i \setminus \{+\textsc{Dorsal}\}) \sqcup \{-\textsc{Dorsal}\}$$
$$\textsc{Cnd} = \mathcal{N}(\{-\textsc{Back}\})$$

captures these alternations.

Case: /łeło/ 'goal' ↦ [leło]. *Search.* Let $w = \langle$ł, e, ł, o$\rangle$. Laterals s_1 and s_3 are in Inr; with $\textsc{Trm} = \mathcal{N}(\emptyset)$, every segment qualifies as a terminator, so search always stops at the segment immediately adjacent to the search initiator. Searching rightward: $\Theta_R(1) = 2$ (terminator s_2 = e) and $\Theta_R(3) = 4$ (terminator s_4 = o).

Change. Since s_2 = e $\in$ Cnd, position 1 is licensed and $s'_1 = \textsc{Out}(s_1, s_2) =$ l. Since s_4 = o $\notin$ Cnd, position 3 is not licensed and $s'_3 = s_3 =$ ł.

Result. /łeło/ ↦ [leło]. The second lateral is unchanged because its right neighbor is a +Back vowel, which does not satisfy Cnd; forms like /xoło/ surface unchanged for the same reason.

A subset of the Kenstowicz and Kisseberth (1979) dataset solved using R_1 is shown below. The first table gives four forms where the rule applies.

UR	/łeło/	/xeli/	/kłeba/	/cʰoli/
R_1	[leło]	[xeli]	[kleba]	[cʰoli]
SR	[leło]	[xeli]	[kleba]	[cʰoli]

These forms show the expected alternation of dark [ɬ] to clear [l] when the rule's local environment is satisfied.

The next table gives four forms where R_1 does not apply.

UR	/kała/	/xoło/	/łxena/	/kbiłs/
R_1	—	—	—	—
SR	[kała]	[xoło]	[łxena]	[kbiłs]

These forms confirm that outside that environment, the laterals surface unchanged.

4.2 Votic Vowel Harmony (Long-Distance)

Leduc et al. (2020) analyze suffix vowels in Votic as underspecified for Back, receiving a value from the nearest preceding non-/i/ vowel. Root vowels are fully specified; suffix vowels are underlyingly underspecified for Back and surface as one of the following alternants:

$$[\text{y}] \sim [\text{u}] \qquad [\text{e}] \sim [\text{ə}] \qquad [\text{æ}] \sim [\text{ɑ}]$$

Votic's inventory of fully specified vowels is:

	i	y	ɨ	u	e	ø	ə	o	æ	ɑ
High	+	+	+	+	−	−	−	−	−	−
Low	−	−	−	−	−	−	−	−	+	+
Back	−	−	+	+	−	−	+	+	−	+
Round	−	+	−	+	−	+	−	+	−	−

The following suffix vowels are underspecified for Back and correspond to the three alternation pairs seen earlier:

$$/\text{U}/ = \{+\textsc{High}, -\textsc{Low}, +\textsc{Round}\}$$
$$/\text{E}/ = \{-\textsc{High}, -\textsc{Low}, -\textsc{Round}\}$$
$$/\text{A}/ = \{-\textsc{High}, +\textsc{Low}, -\textsc{Round}\}$$

The resulting surface alternant is then determined by the Back value the suffix vowel receives.

A challenge noted by Leduc et al. (2020) is that "any vowel except /i/" does not form a natural class (Gorman and Reiss, 2025b). Formally, every natural class defined from a projection of /i/ contains at least one other segment:

$$\forall F \subseteq \mathcal{F}, \quad \mathcal{N}(\pi_F(\mathrm{i})) \neq \{\mathrm{i}\}.$$

To avoid this problem, Leduc et al. (2020) propose two copying rules plus a default rule:

$$F_{\text{NON-HIGH}} = \{+\textsc{Syllabic}, -\textsc{High}\}$$
$$F_{\text{HIGH-ROUND}} = \{+\textsc{Syllabic}, +\textsc{High}, +\textsc{Round}\}$$

R_2 copies Back from the nearest preceding non-high vowel:

$$R_2 = \langle \textsc{Inr}, \textsc{Trm}, \textsc{Dir}, \textsc{Out}, \textsc{Cnd} \rangle$$
$$\textsc{Inr} = \mathcal{N}(\{+\textsc{Syllabic}\})$$
$$\textsc{Trm} = \mathcal{N}(F_{\text{NON-HIGH}})$$
$$\textsc{Dir} = \textsc{Left}$$
$$\textsc{Out} = \lambda s_i.\, \lambda s_j.\, s_i \sqcup \pi_{\{\textsc{Back}\}}(s_j)$$
$$\textsc{Cnd} = \mathcal{N}(\emptyset)$$

R_3 and R_4 share Inr, Dir, and Cnd with R_2, differing only in Trm and Out:

$$R_3: \quad \textsc{Trm} = \mathcal{N}(F_{\text{HIGH-ROUND}})$$
$$\textsc{Out} = \lambda s_i.\, \lambda s_j.\, s_i \sqcup \pi_{\{\textsc{Back}\}}(s_j)$$
$$R_4: \quad \textsc{Trm} = \mathcal{N}(\emptyset)$$
$$\textsc{Out} = \lambda s_i.\, \lambda s_j.\, s_i \sqcup \{-\textsc{Back}\}$$

Note that here, the target class is intentionally broad: all syllabic segments are potential targets, but fully specified vowels remain unchanged, since adding a feature value to a fully specified segment through unification applies vacuously (Bale et al., 2014).

Applied in order, $R_2 \to R_3 \to R_4$ reproduces the full harmony pattern.

Case: /puhE/ 'tree.ILLAT' ↦ [puhə]. *Search.* Consider the word $w = \langle \mathrm{p}, \mathrm{u}, \mathrm{h}, \mathrm{E} \rangle$. The segments $s_2 = \mathrm{u}$ and $s_4 = \mathrm{E}$ are both members of $\textsc{Inr} = \mathcal{N}(\{+\textsc{Syllabic}\})$. Under R_3 ($\textsc{Trm} = \mathcal{N}(F_{\text{HIGH-ROUND}})$, $\textsc{Dir} = \textsc{Left}$):

1. $\Theta_L(2)$ is undefined (no high-round vowel precedes s_2).
2. $\Theta_L(4) = 2$ (terminator is $s_2 = \mathrm{u}$, which is a high-round vowel).

Change. $\textsc{Cnd} = \mathcal{N}(\emptyset)$, so position 4 is licensed.

1. $s'_4 = \textsc{Out}(s_4, s_2) = \mathrm{E} \sqcup \pi_{\{\textsc{Back}\}}(\mathrm{u}) = \mathrm{ə}$.

/E/ receives +Back from the preceding /u/, surfacing as [ə]. R_2 does not apply (no non-high vowel precedes /E/); R_3 bleeds R_4 by filling the underspecified vowel first, so R_4 applies only vacuously, since the relevant segment has already been specified as +Back at that point in the derivation.

Result. $\langle \mathrm{p}, \mathrm{u}, \mathrm{h}, \mathrm{E} \rangle \mapsto \langle \mathrm{p}, \mathrm{u}, \mathrm{h}, \mathrm{ə} \rangle$, i.e., /puhE/ surfaces as [puhə].

A subset of the dataset from Leduc et al. (2020) is shown below, illustrating the three-rule ordering.

UR	/vəttimisE/	/puhE/	/siliA/	/tyttærikkoA/
R_2	[vəttimisə]	—	—	[tyttærikkoɑ]
R_3	—	[puhə]	—	—
R_4	—	—	[siliæ]	—
SR	[vəttimisə]	[puhə]	[siliæ]	[tyttærikkoɑ]

The next section presents an explicit procedure for efficiently computing these mappings.

5 Algorithmic Implementation of Search & Change

The naive approach of evaluating the specification at each position requires a linear scan per position to compute $\Theta_R(i)$ or $\Theta_L(i)$, yielding quadratic worst case time complexity. The key observation is that many search initiating segments can share the same search terminating segment, so once a licensed terminator is found, it licenses all initiating segments in its scope. This enables a linear-time implementation of S&C computable in a single scan of the input string. This paper restricts attention to feature-changing and filling processes, for which no explicit word boundary symbols are required; boundary-sensitive processes, which require additional machinery, are left to future work.

5.1 Reverse Scan with Single Pointer

Scan the word in the direction opposite Dir, maintaining a licensor index j (initialized to -1). At each position k: first compute the output (apply Out if $s_k \in \textsc{Inr}$ and $j \neq -1$, otherwise copy s_k), and only then update j: set $j \leftarrow k$ if $s_k \in \textsc{Trm} \cap \textsc{Cnd}$, reset $j \leftarrow -1$ if $s_k \in \textsc{Trm} \setminus \textsc{Cnd}$, and leave j unchanged otherwise. The output step uses the value of j inherited from the previous position; updating j first would incorrectly allow s_k to license itself. For $\textsc{Dir} = \textsc{Left}$, we reverse the input first and reverse the output after. This follows the same general reversal insight used in harmony work by Heinz and Lai (2013).

The algorithm runs in linear time with only constant-time work per position. Its single integer

Input: word $w = \langle s_1, \ldots, s_n \rangle$, rule $R = \langle \textsc{Inr}, \textsc{Trm}, \textsc{Dir}, \textsc{Out}, \textsc{Cnd} \rangle$
Output: output word w'

if $\textsc{Dir} = \textsc{Left}$ **then**
  $w \leftarrow \textsc{Reverse}(w)$
$w' \leftarrow \epsilon$
$j \leftarrow -1$
for $k \leftarrow n$ **to** 1 **do**
  if $s_k \in \textsc{Inr} \wedge j \neq -1$ **then**
    $s'_k \leftarrow \textsc{Out}(s_k, s_j)$
  else
    $s'_k \leftarrow s_k$
  $w' \leftarrow s'_k \cdot w'$
  if $s_k \in \textsc{Trm} \wedge s_k \in \textsc{Cnd}$ **then**
    $j \leftarrow k$
  else if $s_k \in \textsc{Trm} \wedge s_k \notin \textsc{Cnd}$ **then**
    $j \leftarrow -1$
if $\textsc{Dir} = \textsc{Left}$ **then**
  $w' \leftarrow \textsc{Reverse}(w')$
return w'

Algorithm 1: Reverse Scan with Single Pointer

state makes compilation to finite-state transducers straightforward, as shown in Section 6.

Theorem 1. *Algorithm 1 computes exactly the mapping defined by the S&C specification (Equation 6).*

Proof. It suffices to prove correctness for $\textsc{Dir} = \textsc{Right}$; the $\textsc{Dir} = \textsc{Left}$ case follows by the reversal reduction. Define for each position i:

$$\textsc{Lic}(i) \iff \Theta_R(i)\downarrow \wedge\, s_{\Theta_R(i)} \in \textsc{Cnd},$$

$$\textsc{Idx}(i) = \begin{cases} \Theta_R(i) & \text{if } \textsc{Lic}(i) \\ -1 & \text{otherwise.} \end{cases}$$

We scan from $k = n$ down to 1, maintaining invariants (I1) $\forall m > k,\ s_m^{\text{alg}} = s_m^{\text{spec}}$ and (I2) $j = \textsc{Idx}(k)$.

Base case ($k = n$): no terminator lies strictly to the right of n, so $\textsc{Idx}(n) = -1$. The algorithm initializes $j \leftarrow -1$, so (I2) holds. (I1) is vacuously true.

Inductive step. Assume (I1) and (I2) hold before processing k. For the output at k: the algorithm applies $\textsc{Out}$ only when $s_k \in \textsc{Inr} \wedge j \neq -1$, which by (I2) is equivalent to $s_k \in \textsc{Inr} \wedge \textsc{Lic}(k)$, which is exactly the licensed case of Equation 6. In that case $j = \Theta_R(k)$ and the algorithm writes $s_k^{\text{alg}} = \textsc{Out}(s_k, s_{\Theta_R(k)})$, which matches the specification; otherwise $s_k^{\text{alg}} = s_k$. So (I1) is preserved. For j (to satisfy (I2) at $k-1$): if $s_k \in \textsc{Trm} \cap \textsc{Cnd}$, then $\textsc{Idx}(k-1) = k$ and the algorithm sets $j \leftarrow k$; if $s_k \in \textsc{Trm} \setminus \textsc{Cnd}$, then $\Theta_R(k-1) = k$ (since $s_k \in \textsc{Trm}$ is the nearest terminator), but $s_k \notin \textsc{Cnd}$ makes $\textsc{Lic}(k-1)$ false, so $\textsc{Idx}(k-1) = -1$. Every $i < k$ whose nearest terminator is also k is also unlicensed, so the algorithm correctly carries $j \leftarrow -1$ leftward until the next $\textsc{Trm}$ segment is reached; if $s_k \notin \textsc{Trm}$, then $\textsc{Idx}(k-1) = \textsc{Idx}(k)$ and j is unchanged (since k is not a candidate terminator, $\Theta_R(k-1)$ skips it and returns the same nearest terminator as $\Theta_R(k)$). Therefore (I2) is preserved. By induction, $w^{\text{alg}} = w^{\text{spec}}$. □

6 Compiling S&C Rules into Finite-State Transducers

Algorithm 1 is not itself a finite-state operation, as the reversal step requires reading the full input string before producing output, which is incompatible with streaming computation. What maps onto an FST is not the algorithm as written, but the control structure that it exposes: a single integer state j and a case split on each possible input symbol. The value $j = -1$ corresponds to a single *false* state q_f, and each possible licensor that j can be assigned to corresponds to a *true* state q_σ where $\sigma \in \textsc{Trm} \cap \textsc{Cnd}$. Since control updates and output choices are determined by finitely many (state, input) pairs, all behaviors can be precomputed into transitions. Algorithm 2 gives the full compilation procedure. The compiled transition structure is the same regardless of scan orientation, but its induced transduction depends on how it is traversed: in one orientation it is subsequential, while in the opposite orientation it yields the corresponding reverse-subsequential mapping via conjugation by reversal (Payne, 2017; Heinz, 2018).

Theorem 2. *Let T be the transducer produced by Algorithm 2 with $F_I = \mathcal{F}$. Assuming a fixed rule direction, T computes the same mapping as the corresponding instance of Algorithm 1. The opposite directional mapping is obtained by conjugation with reversal.*

Proof. The transducer states correspond exactly to the original procedure's control values: q_f corresponds to $j = -1$ and q_σ to $j = \sigma$. On input x in state q, both computations emit

$$y = \begin{cases} \textsc{Out}(x, \sigma) & q = q_\sigma \wedge x \in \textsc{Inr} \\ x & \text{otherwise,} \end{cases}$$

Input: feature set $F_I \subseteq \mathcal{F}$, rule $R = \langle \textsc{Inr}, \textsc{Trm}, \textsc{Dir}, \textsc{Out}, \textsc{Cnd} \rangle$ over $\Omega(F_I)$
Output: transition structure $T = \langle Q, \Sigma, \Gamma, I, F, \delta \rangle$ whose induced transduction depends on scan orientation

$\Sigma \leftarrow \Omega(F_I)$
$\Gamma \leftarrow \Omega(F_I)$
$Q \leftarrow \{q_f\} \cup \{q_\sigma \mid \sigma \in \textsc{Trm} \cap \textsc{Cnd}\}$
$I \leftarrow \{q_f\}$
$F \leftarrow Q$
$\delta \leftarrow \emptyset$
foreach $x \in \Sigma$ **do**
  if $x \in \textit{Trm} \cap \textit{Cnd}$ **then**
    $\delta(q_f, x) \leftarrow (q_x, x)$
  else
    $\delta(q_f, x) \leftarrow (q_f, x)$
foreach $\sigma \in \textit{Trm} \cap \textit{Cnd}$ **do**
  foreach $x \in \Sigma$ **do**
    if $x \in \textit{Inr}$ **then**
      $y \leftarrow \textsc{Out}(x, \sigma)$
    else
      $y \leftarrow x$
    if $x \in \textit{Trm} \cap \textit{Cnd}$ **then**
      $\delta(q_\sigma, x) \leftarrow (q_x, y)$
    else if $x \in \textit{Trm} \setminus \textit{Cnd}$ **then**
      $\delta(q_\sigma, x) \leftarrow (q_f, y)$
    else
      $\delta(q_\sigma, x) \leftarrow (q_\sigma, y)$
return $\langle Q, \Sigma, \Gamma, I, F, \delta \rangle$

Algorithm 2: Compilation of an S&C Rule into an FST

and update the control value by

$$q \leftarrow \begin{cases} q_x & x \in \textsc{Trm} \cap \textsc{Cnd} \\ q_f & x \in \textsc{Trm} \setminus \textsc{Cnd} \\ q & x \notin \textsc{Trm}. \end{cases}$$

Algorithm 2 defines exactly one transition per (state, input) pair encoding this case split. By induction on prefix length (base case: both start in q_f; inductive step: both case splits coincide at each position), state and output agree throughout. Hence both computations agree on all strings in $\Omega(\mathcal{F})^*$. □

6.1 Space Complexity

If Trm and Cnd are defined by consistent bundles fixing a set $V_{TC} \subseteq \mathcal{F}$ of features, then $|\textsc{Trm} \cap \textsc{Cnd}| = 3^{|\mathcal{F}|-|V_{TC}|}$, giving

$$|Q| = 1 + 3^{|\mathcal{F}|-|V_{TC}|}, \qquad |\delta| = |Q| \cdot 3^{|\mathcal{F}|}.$$

This is finite in principle but infeasible at realistic scales. For example, with $|\mathcal{F}| = 30$ and $|V_{TC}| = 5$:

$$|Q| = 1 + 3^{25} = 847{,}288{,}609{,}444,$$
$$|\delta| = |Q| \cdot 3^{30} \approx 1.74 \times 10^{26}.$$

The next subsection shows how to compress these generated FSTs by retaining only those state and transition distinctions that are relevant to the rule's behavior.

6.2 Compressing Generated FSTs

Rules interact with only a subset $V \subseteq \mathcal{F}$ of features, where $V = V_{\text{cls}} \cup V_{\text{out}}$: features referenced in Inr, Trm, Cnd and those referenced inside Out. Compiling over the restricted alphabet

$$\Sigma_V = \{\, \pi_V(x) \mid x \in \Omega(\mathcal{F}) \,\}, \quad |\Sigma_V| = 3^{|V|},$$

yields a transducer $T_V : \Sigma_V^* \to \Sigma_V^*$.

Canonical representatives. Let the canonicalization map be

$$\kappa : \Omega(\mathcal{F}) \to \Sigma_V \quad \text{where} \quad \kappa(x) = \pi_V(x)\,.$$

The map κ identifies each segment with its equivalence class under indistinguishability with respect to the rule's visible feature set V. Overload κ to strings by

$$\kappa : \Omega(\mathcal{F})^* \to \Sigma_V^*$$
$$\kappa(\langle x_1, \ldots, x_n \rangle) = \langle \pi_V(x_1), \ldots, \pi_V(x_n) \rangle\,.$$

The pipeline is: project the input to V-features, transduce, then restore the non-V features from the original. Then

$$f(w) = R(T_V(\kappa(w)),\, w)\,,$$
$$R(T_V(\kappa(w)), w)_i = (w_i \setminus \kappa(w)_i) \cup T_V(\kappa(w))_i,$$

where R replaces the V-visible features of each input segment with those produced by T_V, leaving all non-V features untouched. The compressed state and arc counts are

$$|Q'| = 1 + 3^{|V|-|V_{TC}|}, \qquad |\delta'| = |Q'| \cdot 3^{|V|}.$$

Since $|V|$ is much smaller than $|\mathcal{F}|$ in typical analyses, this reduction can be dramatic. For the Votic harmony rule that copies Back from a preceding high-round vowel, the visible features are $V = \{\textsc{Syllabic}, \textsc{High}, \textsc{Round}, \textsc{Back}\}$, so $|V| = 4$ and $|\Sigma_V| = 3^4 = 81$. Since Trm fixes Syllabic, High, and Round, we have $|V_{TC}| = 3$, which yields just $|Q'| = 1 + 3^{4-3} = 4$ states and $|\delta'| = 4 \cdot 81 = 324$ arcs. The correctness of this pipeline is established by the following theorem.

Theorem 3. *The visible-feature pipeline* $f(w) = R(T_V(\kappa(w)), w)$ *computes the same mapping as the full S&C specification.*

Proof. First, since every feature referenced by INR, TRM, and CND lies in V, projection to V preserves all three membership tests: $\pi_V(x) = \pi_V(y)$ implies x and y agree on all valued features with base feature in V, so $x \in \mathcal{N}(A) \iff y \in \mathcal{N}(A)$ for any $A \subseteq (\{+,-\} \times V)$. Second, by Theorem 2 instantiated with feature set V, the transducer T_V computes the projected S&C mapping on Σ_V^*: it applies $\textsc{Out}(x, \sigma)$ at every licensed position and copies elsewhere, with the same control updates as the single-pass procedure over V-projected symbols. Since OUT references only features in V, this projected output is exactly $\pi_V(s'_i)$ at each position i. Third, reconstruction $R(T_V(\kappa(w)), w)_i = (w_i \setminus \kappa(w)_i) \cup T_V(\kappa(w))_i$ re-inserts the non-visible features from the original input, yielding the full output segment s'_i at every position. □

Corollary 4 (Implicit State Merging Under Projection)**.** *Compiling with feature set V rather than $\mathcal{F}$ implicitly merges states into V-projection equivalence classes. Since Algorithm 2 builds over $\Omega(F_I)$ for its given F_I, setting $F_I = V$ causes all membership tests for INR, TRM, and CND to be evaluated only over V-features, collapsing V-equivalent states without changing the realized transduction.*

6.3 Subsequentiality

For every state q and input symbol x, the compiler defines exactly one transition $\delta(q, x)$, making the resulting transducers sequential in the sense of Mohri (1997). Since every sequential transducer is a special case of a subsequential transducer (with empty final outputs), S&C rules compile to subsequential FSTs for a fixed scan orientation.

An important distinction to make is the one between ordinary left-to-right subsequentiality and right-to-left, or reverse, subsequentiality: a mapping is reverse-subsequential when it is realized by reversing the input, applying a subsequential transducer, and reversing the output back (Heinz, 2018; Payne, 2017). For the opposite orientation, the same transition structure induces the corresponding reverse-subsequential mapping by exactly this conjugation with reversal. Thus, the compiler builds a single transition structure, while the choice of scan orientation determines which induced transduction it realizes.

7 Conclusion

Previous work has shown that phonological patterns can be modeled using finite-state transducers (Kaplan and Kay, 1994; Mohri and Sproat, 1996; Mohri, 1997). Unlike subsequential transducers defined directly over string alphabets, S&C rules are stated at the level of phonological features and natural classes, making the linguistic motivation of each rule transparent and independently evaluable. This paper shows that these two perspectives are compatible: S&C rules can be compiled via an explicit construction procedure into a single transition structure whose induced transduction depends on scan orientation: in one orientation it is subsequential, and in the opposite orientation it yields the corresponding reverse-subsequential mapping via conjugation by reversal. Two broader consequences follow from these results.

Learnability. Transductions expressible by subsequential FSTs are learnable from positive input/output string pairs (Oncina et al., 1993; Heinz, 2018), in the sense of identification in the limit (Gold, 1967). S&C-defined mappings therefore fall within a learnable profile: for one orientation they are subsequential, and for the other they are obtained by conjugating a subsequential mapping with reversal. This suggests that the hypothesis-class complexity of S&C-definable mappings can be bounded by restricting learners to subsequential transductions together with reversal, rather than arbitrary string-to-string functions.

Algebraic classification. Regular languages admit classification into algebraic and logical subclasses (Straubing, 1994; Heinz, 2018), and analogous methods exist for deterministic FSTs (Lambert and Heinz, 2023). Since S&C rules compile to FSTs, S&C-modelable mappings can be situated within these classes. A natural next step is a typology of S&C rule schemas grounded in transducer classification, which would yield testable predictions about the computational profiles of phonological processes. Tools such as LTK (Lambert, 2026) could be used to examine which rule families compile to which FST classes systematically.

References

Alan Bale, Maxime Papillon, and Charles Reiss. 2014. Targeting underspecified segments: A formal analysis of feature-changing and feature-filling rules. *Lingua*, 148:240–253.

Alan Bale and Charles Reiss. 2015. *Phonology: A formal introduction*. The MIT Press.

Alan Bale, Charles Reiss, and David Ta-Chun Shen. 2019. Sets, rules and natural classes: [ ] vs. { }. *Loquens*, 6(2):e065.

Noam Chomsky and Morris Halle. 1968. *The Sound Pattern of English*. Harper & Row. Simultaneous rule application discussion on p. 344.

Edgar F. Codd. 1970. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387.

Rim Dabbous, Marjorie Leduc, Fatemeh Mousavi, Charles Reiss, Ta-Chun David, and Shen. 2021. Satisfying long-distance relationships (without tiers): A strictly anti-local approach to phonology. Unpublished manuscript.

Rim Dabbous, Marjorie Leduc, Charles Reiss, and David Shen. 2024. Locality is epiphenomenal: Adjacency is opaqueness. In *Proceedings of the Canadian Linguistics Association 2024*, Carleton University, Ottawa. Conference presentation.

E. Mark Gold. 1967. Language identification in the limit. *Information and Computation*, 10(5):447–474.

Kyle Gorman and Charles Reiss. 2025a. Metaphony in substance free logical phonology. To appear in a special issue of *Phonology*. LingBuzz/008634 (v3, October 2025).

Kyle Gorman and Charles Reiss. 2025b. Natural class reasoning in segment deletion rules. Paper presented at NELS 56, to appear in the proceedings. LingBuzz/009501 (November 2025).

Jeffrey Heinz. 2018. *The Computational Nature of Phonological Generalizations*.

Jeffrey Heinz and William J. Idsardi. 2017. Computational phonology today. *Phonology*, 34(2):211–219.

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.

Ronald M. Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

Michael Kenstowicz and Charles Kisseberth. 1979. *Generative Phonology: Description and Theory*. Academic Press, New York. Simultaneous rule application discussion on p. 318.

Dakotah Lambert. 2026. The language toolkit. Computer software.

Dakotah Lambert and Jeffrey Heinz. 2023. An algebraic characterization of total input strictly local functions. *Proceedings of the Society for Computation in Linguistics*, 6(1):25–34.

Marjorie Leduc, Charles Reiss, and Veno Volenec. 2020. Votic vowel harmony in substance free logical phonology. Published in *The Oxford Handbook of Vowel Harmony*. LingBuzz/005339.

Mehryar Mohri. 1997. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311.

Mehryar Mohri and Richard Sproat. 1996. An efficient compiler for weighted rewrite rules. *Preprint*, arXiv:cmp-lg/9606026.

Jose Oncina, Pedro Garcia, and Enrique Vidal. 1993. Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):448–458.

Amanda Payne. 2017. All dissimilation is computationally subsequential. *Language*, 93(4):e353–e371.

Charles Reiss. 2021. Towards a complete logical phonology model of intrasegmental changes. *Glossa: A Journal of General Linguistics*, 6(1):107.

Charles Reiss. 2022. Priority union and feature logic in phonology. *Linguistic Inquiry*, 53(1):199–209.

Charles Reiss and Veno Volenec. 2022. Conquer primal fear: Phonological features are innate and substance-free. *Canadian Journal of Linguistics / Revue canadienne de linguistique*, 67(4):581–610.

Howard Straubing. 1994. *Finite Automata, Formal Logic, and Circuit Complexity*. Birkhäuser.

What Matters in Tonotactic Learning

Han Li and Jeffrey Heinz
Stony Brook University
Department of Linguistics
Institute for Advanced Computational Science
{han.li.4,jeffrey.heinz}@stonybrook.edu

Abstract

This paper investigates whether tonotactic learning differs across representations and learning models. We conduct an experiment using the same dataset encoded in three representations: segments, features, and autosegmental representations (ARs). To the extent possible, two learning models are evaluated, the Maximum Entropy (MaxEnt) model and the Bottom-Up Factor Inference Algorithm (BUFIA), to examine how learning outcomes interact with both model type and representations. A follow-up experiment further explores the roles of frequency and complexity thresholds. The results show that (1) AR-based learning gives the strongest overall performance; (2) there is no consistent advantage between segmental and feature-based representations across learning models; (3) MaxEnt performance improves substantially when frequency information is introduced and lastly (4) the effects of complexity bounds interact with representation type and frequency information. These findings suggest that tonotactic learning benefits from structurally explicit representations. Overall this work highlights the importance of using linguistically meaningful representations in learning.

1 Introduction

This paper focuses on the learnability of tonal phonotactics, or tonotactics, which addresses the question of how tones and tone-bearing units (TBUs, such as syllable or mora) are organized in tonal languages. More specifically, we examine how the learning outcomes are influenced by representations and learning models.

For a long time, despite the widespread distribution of tonal languages across the world (Yip, 2002), tonotactics has received considerably less attention in computational learning than segmental phonotactics. One possible reason for this gap lies in representational differences. Segmental phonotactics is typically modeled using linear representations such as strings, which makes it readily compatible with a wide range of existing learning models. (Coleman and Pierrehumbert, 1997; Hayes and Wilson, 2008; Goldsmith and Riggle, 2012; Mayer and Nelson, 2020; Schrimpf and Jarosz, 2014). Tonotactics, by contrast, has long been analyzed over non-linear structures in linguistics, most widely autosegmental representations (ARs, Goldsmith, 1976). These more complex representations pose challenges for many existing computational models, which are unable to operate directly over non-linear structures. As a workaround, many approaches usually flatten tonal patterns into strings to fit linear learning frameworks (Shih and Inkelas, 2013). Although a substantial body of work has argued that linear representations are inadequate for capturing tonal generalizations (Clements and Goldsmith, 1984; Odden, 2013) and examined the computational properties of ARs (Bird and Klein, 1990; Coleman and Local, 1991; Kornai, 2007; Jardine and Heinz, 2015; Jardine, 2016, 2017, 2019), we are unaware of any empirical demonstrations of clear advantages for non-linear representations over strings in tonotactic learning.

To fill this gap, we conducted computational experiments and encoded the same dataset in three different representational systems commonly used in tonal phonology, namely strings, sequences of features, and ARs. The string (or segmental) representation uses a sequence of symbols representing low (L), high (H), rising (R), and falling (F) tones. The feature-based representation encodes the data as a sequence of feature bundles with the features [±high] and [±contour]. The third representation provides ARs, in which tones and tone-bearing units are represented on separate tiers linked by association relations (see §2).

These encodings are grounded in theoretical phonology, and differ in the degree to which tone is independent from the TBUs. Since this study fo-

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 180–190
July 3-4, 2026 ©2026 Association for Computational Linguistics

cuses on the effect of representational differences on phonotactic learning, we assume that representations are given to the learner rather than learned from the data. Different representations are evaluated using two learning models: the Bottom-Up Factor Inference Algorithm (BUFIA; Chandlee et al., 2019) and the Maximum Entropy phonotactic learner (MaxEnt; Hayes and Wilson, 2008). BUFIA is a categorical learner that induces a set of inviolable constraints, whereas MaxEnt is a gradient learner that assigns probabilistic weights to constraints. In addition, a set of linguistically motivated grammars, encoded in the corresponding representations, serves as a baseline for comparison with the learned grammars.

One limitation of this study is that the current implementation of MaxEnt does not support ARs. A more comprehensive study would also include a version of MaxEnt which operates over ARs. In addition, neural models have been proposed for learning phonotactic or stress patterns (Goldsmith, 1994; Mayer and Nelson, 2020; Prickett and Pater, 2025; Gupta and Touretzky, 1991), which are not studied here. Furthermore, we do not address the problem of representational parsing or acquisition.[1] We leave these questions for future work. Despite these limitations, the results presented here help assess the extent to which ARs facilitate learning.

The corpus in this study comes from the Hausa database in the *World Loanword Database* (Awagana et al., 2009), which was curated by Li (2025). The corpus contains 621 mono-morphemic, non-borrowed words (64 types) from Hausa's core vocabulary, transcribed in orthography.

This corpus was randomly divided into four partitions (folds). Each fold was converted into the three representational schemes. The tonal representations studied here abstract away from segmental content and retain only tonal and TBU information. For each representation and model, cross-fold validation was conducted and their model performance was examined. In every case, the evaluation set was augmented to include forms that violated the constraints in the baseline grammar, and precision, recall, and F1 scores were reported. These measures help identify whether the models generalize appropriately without overgeneralizing.

The results show three main findings. First, segmental and feature-based representations do not show a consistent difference in learning outcomes: they perform more or less identically under BUFIA and only show marginal and inconsistent differences under MaxEnt. Second, BUFIA over ARs achieves the best performance. Third, learning outcomes are influenced by other factors such as frequency information, complexity bounds, and evaluation criteria. Frequency substantially improves MaxEnt performance and gives comparable results to BUFIA-AR, while categorical learners such as BUFIA remain stable without frequency.

Overall, this work highlights the importance of incorporating linguistically meaningful representations into computational models of phonological learning. It further suggests these approaches should be evaluated on larger-scale and more complex datasets to better identify differences between the approaches.

2 Background

This section provides the theoretical and computational background and methods for the different representations investigated here. We adopt a model-theoretic perspective that treats all representations as different types of mathematical objects (Enderton, 2001), or more specifically, relational structures (Rogers et al., 2013; Chandlee et al., 2019; Rogers and Lambert, 2019).

Model-theoretic representations not only provide a unified framework for encoding different representational systems, but also make explicit how linguistic units are structurally related to one another. One can observe that the more enriched representations encode more complex dependency relations, which in turn lead to different generalizations for phonotactic learners.

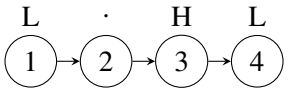
2.1 Representations

We define a phonological representation as relational structures of the form $\langle D, \mathcal{R} \rangle$, where D is a finite domain of positions, and $\mathcal{R}$ is a set of relations (also known as the model *signature*) that connect the composing units of the phonological representation (Strother-Garcia et al., 2017; Jardine, 2016). Relations may be unary, indicating the identity of an element, or binary, specifying the relationship between pairs of elements.

The string representation arranges symbols along a single axis. The signature of a string model includes unary labeling relations indicating

[1] Though see Jardine and Heinz (2015) for one method for deriving ARs from strings.

the symbols, and a single binary successor relation $\rightarrow$ defining linear order. For example, the string representation of the tonal structure of the Hausa word *mù.tûm* ("person") is L.HL. It is visualized in Figure 1. The domain $D = \{1, 2, 3, 4\}$ specifies four positions, which are labeled by their tone and syllable the boundary marker (.). The successor relation encodes the temporal order. Each position occupied by a tone can be thought of representing a single mora. Thus bimoraic syllables will contain a sequence of two tones. For complex tonal forms where a single mora is associated with contour tones, the symbols R (representing rising LH on one mora) and F (representing falling HL on one mora) are used.

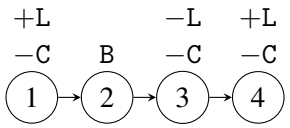


$$
\begin{aligned}
D &= \{1, 2, 3, 4\} \\
L &= \{1, 4\} \\
H &= \{3\} \\
. &= \{2\} \\
R &= \varnothing \\
F &= \varnothing \\
\rightarrow &= \{\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 3, 4\rangle\}
\end{aligned}
$$

Figure 1: String model when $\Sigma = \{H, L, R, F, .\}$ for the Hausa word *mù.tûm* ("person")

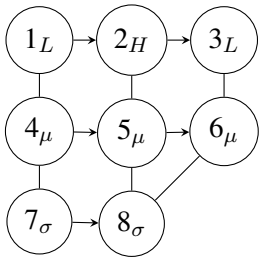
The feature-based representation also arranges symbols along a single axis. Like the string model, the signature of the feature model includes single binary successor relation $\rightarrow$ defining linear order. Unlike the string models, there are five unary relations: `{+Low, -Low, +Contour, -Contour, Boundary}`, where `low` and `contour` are binary features, and `Boundary` is treated as a privative feature. In the string model, each position is true of exactly one unary relation. However, in the feature model, positions may be true of more than one unary relation. Strother-Garcia et al. (2017) and Vu et al. (2018) refer to such models as *unconventional* because they are not the standard way to model strings in theoretical computer science (though it is standard in phonology). Note in this representation `[±Contour]` is only used to indicate monomoraic contour tones. Figure 2 shows the feature representation of same tonal structure of the word in Figure 1.

ARs, in contrast, correspond to multi-linear (tiered) relational structures. The tiers correspond to units such as tones, moras, and syllables. The unary relations are {L, H, μ, σ}, denoting low and high tones, moras and syllables respectively. The signature not only includes the successor relation, which relates positions on the same tier, but it also includes a binary association relation α. This relation captures associations across tiers. In this representation, structural dependencies, such as tone spreading or contour formation, are encoded directly as relations rather than inferred from symbol sequences. For example, Figure 3 shows the AR model for the same word *mù.tûm*. The model consists of eight units, defined by $D = \{1, \ldots, 8\}$, arranged across three tiers, each following a successor relation $\rightarrow = \{(1, 2), (2, 3), \ldots\}$. The association relations, on the other hand, are expressed as a set of unordered pairs $\alpha = \{(1, 4), (2, 5), \ldots, (6, 8)\}$, indicating which two elements are connected by an association line.



$$
\begin{aligned}
D &= \{1, 2, 3, 4\} \\
\texttt{+Low} &= \{1, 4\} \\
\texttt{−Low} &= \{3\} \\
\texttt{+Contour} &= \varnothing \\
\texttt{−Contour} &= \{1, 3, 4\} \\
\texttt{Boundary} &= \{2\} \\
\rightarrow &= \{\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 3, 4\rangle\}
\end{aligned}
$$

Figure 2: Feature model for *mù.tûm* ("person")



$$
\begin{aligned}
D &= \{1, 2, 3, 4, 5, 6, 7, 8\} \\
L &= \{1, 3\} \quad H = \{2\} \quad \mu = \{4, 5, 6\} \quad \sigma = \{7, 8\} \\
\rightarrow &= \{\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 4, 5\rangle, \langle 5, 6\rangle, \langle 7, 8\rangle\} \\
\alpha &= \{(1, 4), (2, 5), (3, 6), (4, 7), (5, 8), (6, 8)\}
\end{aligned}
$$

Figure 3: Autosegmental model for *mù.tûm* ("person")

This formalization explicitly captures how within-tier and across-tier relations are represented in autosegmental phonology (Goldsmith,

1976). In particular, tonal generalizations have been argued to be computationally simpler over ARs than over strings, because cross-tier dependencies are represented directly as relations. This allows processes to be modeled by modifying associations rather than altering feature values or symbols, and can describe long-distance processes locally (Goldsmith, 1976; Jardine, 2017).

2.2 Phonotactic Learning Models

One learning system used in this paper is the BUFIA (Chandlee et al., 2019) which learns constraints expressed as relational structures under an arbitrary model-theoretic signature. This means the general form of the algorithm can be flexibly applied to very many specific linguistic representations. In this work, we used three versions of BUFIA: BUFIA-Seg, BUFIA-Ftr and BUFIA-AR.

In a nutshell, BUFIA is a deterministic phonotactic learner that traverses a hypothesis space of all logically possible structures, finding those relational structures which are absent in the data. BUFIA constructs the hypothesis space on the fly, beginning from the simplest (smallest) relational structures, and considering increasingly complex ones as needed. The notions 'simpler' and 'more complex' are made concrete by the *containment relation*: some pairs of structures stand in a containment relation (one containing the other, where the more complex structure is referred to as the *superfactor* and the contained structure is the *subfactor*), while others are not comparable. All logically possible structures, which serve as candidate constraints, are partially ordered by the containment relation.

Two core operations of BUFIA determine its search: containment and the sub/superfactor relation. The learner is provided with a set of positive data and a complexity bound. The search begins from an initial structure at the bottom of hypothesis space (typically the empty structure). For each candidate, BUFIA checks whether it is contained in the positive dataset. If it is attested, the learner generates its *next superfactors* (i.e., minimally more complex superfactors) and continues searching upward. Otherwise, the structure is identified as a forbidden factor (i.e., a *constraint*), and its superfactors are pruned from the rest of the space. Chandlee et al. (2019) prove that BUFIA identifies the most general constraints within the complexity bounds that are consistent with the data. The resulting grammar guarantees full coverage of the observed data while remaining maximally general.

As said before, a key merit of BUFIA is its representational flexibility. The algorithm can operate over any representational choices, provided they are defined model-theoretically. Recent work has demonstrated the success of BUFIA in phonotactic learning. Swanson et al. (2026) implemented a string-based version of BUFIA for learning Quechua phonotactics and showed that it performed as well as the (Hayes and Wilson, 2008; Wilson and Gallagher, 2018). Li (2025) introduced a version of BUFIA that operates directly over ARs for tonotactic learning and showed that it could learn tonotactic constraints. This representational flexibility allows controlled comparison of different representations of the same data under a shared learning framework.

BUFIA is a categorical learner: a candidate structure is either well-formed or ill-formed, depending on whether it contains any forbidden subfactor. In contrast, probabilistic models, such as MaxEnt (Hayes and Wilson, 2008), assign gradient well-formedness values to candidate forms. Well-formedness is interpreted as a probability rather than a binary value; consequently, constraints are treated as violable, each associated with a penalty weight. This evaluation is cumulative: the more constraints a candidate violates, and the higher the weights of those constraints, the lower its predicted probability of well-formedness.

MaxEnt has been widely used in string-based phonotactic learning, and more recent implementations induce phonological tiers to capture long-distance patterns (Gouskova and Gallagher, 2020) and metrical structure (Lee et al., 2025). These studies suggest that probabilistic string-based approaches can learn structural information that is not directly visible in surface sequences. However, these representations differ from ARs in that they typically do not encode association relations between individual elements on separate tiers.

The central question of this paper is the extent to which representations facilitate tonotactic learning. Does adopting ARs improve learning outcomes over adopting string and featural representations? A related question is whether learning outcomes are influenced by the type of learner (here BUFIA or MaxEnt). In the following section, we present an experiment designed to test these questions by applying the same dataset across different representations and the two learning schemes.

3 Experiment: Hausa Tonotactics

To test tonotactic learnability across representations and models, we conducted a series of controlled computational experiments on the same dataset. This section details the experimental setup, including construction of the training and test data, and the evaluation metrics used to compare model performance.

3.1 Hausa Data

The language data used in this study come from Hausa, a major tonal language spoken in West Africa. Hausa has three surface tones in orthography: H, L, and HL (a falling tone, which occurs only in heavy syllables). Previously proposed generalizations about Hausa tonotactics are listed in Table 1 (Newman, 2002; Leben, 1996; Zoll, 2003).

Constraints	Interpretation
(1) ***Rise**$_\sigma$	No monosyllabic LH
(2) ***Contour**$_\mu$	No contour on single mora
(3) ***3T**	No trimoraic contour
(4) ***NonFinHL**	No non-final HL
(5) ***Lapse**	No adjacent L-toned syllables

Table 1: Phonological Generalizations in Hausa

We use a curated corpus from Li (2025), constructed from the Hausa database (Awagana et al., 2009) in the *World Loanword Database*. The corpus contains 621 mono-morphemic, non-borrowed words from the core vocabulary, transcribed in orthography. These 621 words yield a total of 64 tonal structures.

Among the constraints in Table 1, the first three constraints, *Rise$_\sigma$,*Contour$_\mu$,*3T, are hard constraints that define the basic phonology of tones and syllable structures of Hausa. They are never violated in the corpus. Constraint *NonFinHL applies to the non-derived Hausa words where the falling contour (HL) can only occur word finally. The constraint *Lapse is proposed by Zoll (2003) to explain the rarity of adjacent L-toned syllables. Newman (2002) points out that there are exceptional words that violate *NonFinHL and *Lapse. Li (2025) also reports that the corpus itself includes forms violating these two constraints.

3.2 Procedure

The experiment compares grammars learned under multiple conditions, varying both learning model (BUFIA vs. MaxEnt) and representation (Segment, Feature, AR). These learned grammars are evaluated against baseline grammars defined over the corresponding representations (Baseline-Seg/Ftr/AR, see Appendix A1). Table 2 summarizes all model conditions. As mentioned, MaxEnt-AR is one model we are unable to test.

Model	length
Baseline-Seg	$k = 3$
Baseline-Ftr	$k = 3$
Baseline-AR	$t = 2, s = 2, m = 3$
BUFIA-Seg	$k = 3$
BUFIA-Ftr	$k = 3$
BUFIA-AR	$t = 2, s = 2, m = 3$
MaxEnt-Seg	$k = 3$
MaxEnt-Ftr	$k = 3$

Table 2: Learning models in the experiment

Both BUFIA and MaxEnt specify thresholds that limit the maximum length or structural complexity of learned constraints. These can be set manually, or "tuned" according to data. However, the data used for tuning cannot be used for training or testing. Since our corpus is small (64 forms), we elected to manually set these thresholds. We chose to set values according to the constraints in the baseline grammar, which helps provide a fair comparison. Thus, we set the thresholds to the most complex constraint in the baseline grammar. For string and featural representations, the most complex baseline constraint has length 3 (e.g., *HL.). For ARs, the threshold consists of three parameters: t (maximum number of tonal units), s (maximum number of syllables), and m (maximum number of moras). Since the most complex constraint in the baseline grammar has two tones, two syllables and three moras, we set $t = 2$, $s = 2$, and $m = 3$.

We evaluated each model using 4-fold cross-validation. The dataset was partitioned into four equally sized folds (16 forms per fold). In each iteration, three folds were used for training and the remaining fold provided the positive items in the test set. This process was repeated four times, with each fold serving as the test set once.

Arguably, if the test set only contains positive items, then it can mislead. For example, a grammar with no constraints will accept the entire test set because no test form violates any constraint. Such a grammar therefore over-generates. The in-

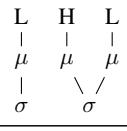
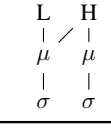
Word	String	Feature	AR
mù.tûm ‘person’	L.HL	L: +0−+ C: −0−− B: 0+0 0	
(nonce)	R.H	L: +0− C: +0− B: 0+0	

Table 3: Positive and negative forms in three representations

clusion of negative forms in the test set helps detect over-generation. While the grammar with no constraints would correctly classify all positive forms, it would also incorrectly classify all negative forms. Similarly, a grammar with too many constraints may correctly classify all negative forms, and incorrectly reject all positive forms. Grammars that more correctly classify both positive and negative forms are desirable.

Therefore, we also constructed negative items to include in the test set. Since there were 16 positive items in each test set, we constructed 16 negative items, and added them to each test set. Thus, each test set contained 32 forms.

In the absence of native speaker judgments, there is less certainty regarding whether an unattested form is negative or positive. For example, the baseline grammar includes two constraints that are known to have exceptions.

We therefore constructed negative forms that met two conditions. First, each negative form was unattested in the corpus. Second, each negative form violated at least one of the hard constraints in Table 1 (constraints 1-3). Since there were 16 negative forms and three constraints, at least five forms violated each constraint. One positive and one negative form are shown in Table 3.

For training and testing each model, all data was converted into each of the three representational systems: string, feature, and ARs. To evaluate model performance, we measured learning outcomes as a binary classification task.

3.3 Evaluation

For grammars returned by BUFIA and the baseline grammars, if a test form violated any of the constraints in the grammar, then the grammar marked it as rejected; otherwise, it was marked as accepted. A prediction was counted as correct if positive forms were accepted and negative forms were rejected. Performance was measured using averaged recall, precision, and F1 score across the folds.

The MaxEnt model produces a gradient grammar with weighted constraints. This grammar assigns a score to each test form based on a weighted sum of constraint violations. A zero score means the form violates no constraints, and higher scores indicate the form violates multiple, or higher-weighted, constraints.

There are many ways this score can be interpreted to produce a binary classification. We considered a simple threshold. If the score the grammar assigns to a form is greater than the threshold then it is rejected; otherwise it is accepted. We examined a few possible thresholds and selected the one that gave the best results. Again, normally the threshold could be set as part of a tuning phase, but we did not want to spare any data for this.

The three thresholds we considered are the following. First, over-zero: a form is rejected if its total weighted violation score is greater than zero. Second, over-average: a form is rejected if its total weighted violation score exceeds the average violation score across all candidate forms. Third, over-median: a form is rejected if its total weighted violation score exceeds the median score across all candidate forms. These criteria are different ways to categorically interpret the gradient grammar.

In all of our experiments, the over-average interpretation performed the best. The over-zero interpretation was the worst, and tended to reject more than half of the positive forms. This is likely because constraints with small non-zero weights were learned. Generally, the over-median approach performed similarly to over-average, but had lower scores. Consequently, in what follows, the MaxEnt results are reported with the over-average threshold.

4 Results

Results of the experiments are shown in Table 4. Several aspects are worth noting. First, the baseline grammars have identical results. A closer inspection of the errors confirms that they also coincide. This indicates a one-to-one correspondence between representations, and the forms that violate these constraints. It follows that representational differences do not affect how this fixed grammar classifies forms.

On the other hand, for the BUFIA and Max-

Model	Recall	Precision	F1
Base-Seg	0.812	1.000	0.894
Base-Ftr	0.812	1.000	0.894
Base-AR	0.812	1.000	0.894
BUFIA-Seg	0.984	1.000	0.992
BUFIA-Ftr	0.984	1.000	0.992
BUFIA-AR	1.000	1.000	1.000
MaxEnt-Seg	0.922	0.827	0.871
MaxEnt-Ftr	0.906	0.773	0.832

Table 4: Average model performance

Ent models, which learn constraints, performance does vary depending on both the representation and the learning model. Among the BUFIA models, while there is no difference between BUFIA-Seg and BUFIA-Ftr, both achieve an F1 score of 0.992. BUFIA-AR, however, performs perfectly (F1 = 1.000). Although it is true that BUFIA-AR outperforms BUFIA-Seg and BUFIA-Ftr, all the BUFIA models are exhibiting ceiling levels of performance, or are extremely close to it.

Choice of representation also impacts the MaxEnt models. MaxEnt-Ftr achieves an F1 score of 0.832, while MaxEnt-Seg reaches 0.871. The better performance on segmental representations over featural ones contrasts with earlier studies with MaxEnt on segmental phonotactics, which found that learned constraints over features were more effective than learned constraints over segments (Hayes and Wilson, 2008; Wilson and Gallagher, 2018).

Finally, the results show that BUFIA-Seg and BUFIA-Ftr outperform MaxEnt-Seg and MaxEnt-Ftr. We were surprised by this result and conducted follow-up experiments to see if we could improve the performance of MaxEnt.

One finding is that ARs provide a learning advantage over segmental or feature-based representations. However, the improvement is marginal as all the BUFIA models performed at, or close to, ceiling. We therefore conclude that while there is some evidence that ARs improve tonotactic learnability, we believe it is premature to draw firm conclusions. The tonotactics of Hausa may simply be too simple to see any significant difference in BUFIA's learning capacities with different representations. Therefore, we think that similar experiments ought to be conducted on languages with more complex tonotactics as that may allow for clearer distinctions to emerge when learning over segmental, featural, or autosegmental representations.

5 Follow-Up Experiments

We were curious what hindered the performance of the MaxEnt model. In the previous experiments, the model was trained on the types rather than tokens; therefore, frequency information was not available during learning. Since MaxEnt determines weights for the constraints it learns partially based on frequencies, we hypothesized their absence may have been a hindrance.

Therefore, in a follow-up experiment, we introduced frequency to MaxEnt by training the model on token data rather than type data (621 tokens in total). As in the previous setup, we conducted cross-fold validation over four folds (each fold contained 156 or 155 tokens). As before, an equal number of negative forms were included in the test set. Since the over-mean criterion gave the best results in the previous experiment, we use it again here to calculate recall, precision, and F1 scores.

Another question is how to determine an appropriate complexity bound that accounts for the data without leading to over-generalization. To examine this, we also ran experiments with MaxEnt models where the maximum constraint length was set to 4. This is the highest value permitted by the implementation we used. We conducted experiments where these MaxEnt models had access to frequency information, and where they did not.

As shown in Table 5, the inclusion of frequency information makes a substantial difference. When frequency information was included, all MaxEnt models performed close to ceiling with F1 scores greater than or equal to 0.990 regardless of the complexity bound. Without frequency information, the highest F1 score, achieved by MaxEnt-Seg, is 0.871 when $k = 3$, and the other MaxEnt models get even lower when k increases to 4.

In terms of the feature vs. segment comparison, the results are inconsistent. When frequency information is not included, the advantage between MaxEnt-Seg and MaxEnt-Ftr reverses between complexity bounds: MaxEnt-Seg performs better at $k = 3$, whereas MaxEnt-Ftr performs better at $k = 4$. Once the frequency information is introduced, there appears to be no appreciable difference among the F1 scores on the grammars obtained by the MaxEnt models between constraint

Model	k	Recall	Precision	F1
MaxEnt-Seg	3	0.922	0.827	0.871
MaxEnt-Ftr	3	0.906	0.773	0.832
MaxEnt-Seg-freq	3	1.000	0.981	0.990
MaxEnt-Ftr-freq	3	1.000	0.981	0.990
MaxEnt-Seg	4	0.644	0.866	0.651
MaxEnt-Ftr	4	0.816	0.844	0.829
MaxEnt-Seg-freq	4	1.000	0.994	0.997
MaxEnt-Ftr-freq	4	1.000	0.980	0.990
BUFIA-Seg/Ftr	3	0.985	1.000	0.992
BUFIA-Seg/Ftr	4	0.969	1.000	0.984
BUFIA-Seg/Ftr	5	0.907	1.000	0.950
BUFIA-Seg/Ftr	6	0.782	1.000	0.873
BUFIA-Seg/Ftr	7	0.656	1.000	0.782

Table 5: Performance of MaxEnt and BUFIA models across complexity bounds and frequency settings. The MaxEnt-Seg and MaxEnt-Ftr results for $k = 3$ are repeated from Table 2 for ease of comparison.

lengths 3 and 4, as these results are effectively at ceiling.

For the sake of comparison, we also examined the grammars returned by the BUFIA models when we varied the maximum sizes of the constraints. For BUFIA-Seg and BUFIA-Ftr, we examined constraint lengths from 3 to 7, as shown in Table 5; For BUFIA-AR, we explored parameter settings for t, m, s up to 3. The results for $m = 3$ are presented in Table 6, while the complete set of results is provided in the Appendix A2.

For BUFIA-Seg and BUFIA-Ftr, the highest F1 score is 0.992 when $k = 3$. For $k = 4$, the F1 is still close to ceiling at 0.984. As k increases, the F1 continues to drop for these models, indicating over-fitting.

For BUFIA-AR, when $m < 3$, the highest F1 score was 0.842. When the search for constraints cannot consider structures with three moras, BUFIA-AR will not find the constraint against trimoraic contours. Similarly, when $t = 1$, the hard constraints against monosyllabic rises and monomoraic contours cannot be learned, which significantly brought down performance. On the other hand, F1 scores were at 1.000 in the following three settings: $[m = 3, s = 1, t = 2], [m = 3, s = 1, t = 3], [m = 3, s = 2, t = 2]$. In the maximum setting when $[m = 3, s = 3, t = 3]$, the F1 score dropped to 0.967, indicating the beginning of the cline towards over-fitting.

m	s	t	Recall	Precision	F1
3	1	1	1.000	0.615	0.762
3	1	2	1.000	1.000	1.000
3	1	3	1.000	1.000	1.000
3	2	1	1.000	0.615	0.762
3	2	2	1.000	1.000	1.000
3	2	3	0.953	1.000	0.975
3	3	1	0.984	0.612	0.754
3	3	2	0.984	1.000	0.992
3	3	3	0.938	1.000	0.967

Table 6: Performance of BUFIA-AR across complexity thresholds when $m = 3$.

6 Discussion and Conclusion

We observe from the results that several factors influence learning outcomes in surface-based tonotactics.

First, Wilson and Gallagher (2018) show that for segmental phonotactics, learning tends to be more successful over feature-based representations than over segments. Our results in tonal phonology, however, do not show a consistent difference between featural and segmental representations. Under the BUFIA model, the two representations yield identical performance, while under MaxEnt the difference is marginal and inconsistent. One possible explanation is that string-based tonal representations are relatively simple, involving a much smaller inventory of tonal symbols (typically two to four tonemes and three to six features) compared with the richer feature systems used in segmental phonology.

In contrast, ARs can be used effectively for learning and consistently outperform string-based representations. This advantage is particularly evident when compared with MaxEnt models without frequency, and the improved performance is marginal when compared with MaxEnt models trained with frequency.

In addition, for gradient learners, frequency information in the dataset is crucial for achieving strong performance; however, this is not required for BUFIA, which achieves competitive results even without frequency information. Furthermore, for a dataset such as Hausa, which exhibits a relatively small set of surface tonal constraints, the optimal complexity threshold appears to lie between $k = 3$ and $k = 5$ for string representations, or under AR settings where $\mu \geq 3$ and $t > 1$. We be-

lieve for more complex tonal systems, learners like BUFIA-AR are advantageous as they allow complexity to be controlled more finely across multiple tiers while searching the hypothesis space.

We aim to draw attention to the use of computational methods for evaluating competing representational choices and theories (Oakden, 2020; Danis, 2025). Models such as BUFIA, which allow flexible representations, can serve as practical tools for facilitating such comparisons. More broadly, computational approaches also allow us to explore how grammar inference can be improved through interactions among representations, token or type data, complexity bounds, and evaluation criteria.

In conclusion, we show that tonotactics can be learned effectively and efficiently over autosegmental representations using BUFIA-AR with sparse and limited data. The model achieves a comparable learning outcome to gradient models that rely heavily on distributional information. The results suggest that AR-based learning is a promising direction for modeling tonal grammars and tonal processes, where phonological changes often involve modifications of associations between tiers rather than individual symbols.

This work remains limited in some respects that motivate future research. First, we conducted the experiments on a small dataset with comparatively simple tonotactic patterns, and therefore the performance across learners does not differ greatly. Future work should evaluate these approaches on larger-scale and more complex datasets. This would enable more statistically meaningful comparisons and experiments on probabilistic or neural phonotactic learners. Nonetheless, we hope that by making non-linear representations available for learning, it will become easier to investigate representational learning itself or phonological transformations in the future.

Acknowledgments

HL would like to express her gratitude to John Goldsmith, Ellen Broselow, Thomas Graf, as well as the audience at the Workshop on The Role of Representation in Computational Phonology, for their valuable feedback and support. HL also gratefully acknowledges the support of the Institute for Advanced Computational Science through the Junior Researcher Award. This research was also supported by US NSF grant #2416183 to JH. The authors would also like to thank the four anonymous reviewers for their thoughtful feedback.

References

Ari Awagana, H. Ekkehard Wolff, and Doris Löhr. 2009. Hausa. In Martin Haspelmath and Uri Tadmor, editors, *World Loanword Database*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Steven Bird and Ewan Klein. 1990. Phonological events. *Journal of Linguistics*, 26(1):33–56.

Jane Chandlee, Rémi Eyraud, Jeffrey Heinz, Adam Jardine, and Jonathan Rawski. 2019. Learning with partially ordered representations. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 91–101.

George N Clements and John Goldsmith. 1984. *Autosegmental studies in Bantu tone*, volume 3. Walter de Gruyter.

J. S. Coleman and J. Pierrehumbert. 1997. Stochastic phonological grammars and acceptability. In *Proceedings of the Third Meeting of the ACL Special Interest Group in Computational Phonology*, pages 49–56. Somerset, NJ: Association for Computational Linguistics.

John Coleman and John Local. 1991. The "No Crossing Constraint" in Autosegmental phonology. *Linguistics and Philosophy*, pages 295–338.

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.

Herbert B Enderton. 2001. *A Mathematical Introduction to Logic*. Elsevier.

John Goldsmith. 1994. A dynamic computational theory of accent systems. In Jennifer Cole and Charles Kisseberth, editors, *Perspectives in Phonology*, pages 1–28. Stanford: Center for the Study of Language and Information.

John Goldsmith and Jason Riggle. 2012. Information theoretic approaches to phonological structure: the case of finnish vowel harmony. *Natural Language & Linguistic Theory*, 30(3):859–896.

John Anton Goldsmith. 1976. *Autosegmental Phonology*. Ph.D. thesis, MIT.

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

Prahlad Gupta and David Touretzky. 1991. What a perceptron reveals about metrical phonology. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pages 334–339.

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

Adam Jardine. 2016. *Locality and non-linear representations in tonal phonology*. Ph.D. thesis.

Adam Jardine. 2017. The local nature of tone-association patterns. *Phonology*, 34(2):363–384.

Adam Jardine. 2019. The expressivity of autosegmental grammars. *Journal of Logic, Language and Information*, 28:9–54.

Adam Jardine and Jeffrey Heinz. 2015. A concatenation operation to derive autosegmental graphs. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, pages 139–151.

András Kornai. 2007. *Mathematical linguistics*. Springer Science & Business Media.

William R Leben. 1996. Tonal feet and the adaptation of english borrowings into hausa. *Studies in African Linguistics*, 25(2):129–154.

Seung Suk Lee, Joe Pater, and Brandon Prickett. 2025. Representing and learning stress in a maxent framework. In *Proceedings of the Annual Meetings on Phonology*, volume 1. University of Massachusetts Amherst Libraries.

Han Li. 2025. Learning tonotactic patterns over autosegmental representations. In *Proceedings of the Annual Meetings on Phonology*, volume 1. University of Massachusetts Amherst Libraries.

Connor Mayer and Max Nelson. 2020. Phonotactic learning with neural language models. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 291–301.

Paul Newman. 2002. *The Hausa Language: An Encyclopedic Reference Grammar*, volume 122. Yale University Press, New Haven, US.

Chris Oakden. 2020. Notational equivalence in tonal geometry. *Phonology*, 37(2):257–296.

David Odden. 2013. *Introducing phonology*. Cambridge University Press.

Brandon Prickett and Joe Pater. 2025. Learning and generalizing stress patterns with a sequence-to-sequence neural network. *Linguistics Vanguard*, (0).

James Rogers, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel. 2013. Cognitive and sub-regular complexity. *Lecture Notes in Computer Science*.

James Rogers and Dakotah Lambert. 2019. Some classes of sets of structures definable without quantifiers. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 63–77, Toronto, Canada. Association for Computational Linguistics.

Natalie M Schrimpf and Gaja Jarosz. 2014. Comparing models of phonotactics for word segmentation. In *Proceedings of the 2014 Joint Meeting of SIGMORPHON and SIGFSM*, pages 19–28.

Stephanie Shih and Sharon Inkelas. 2013. A subsegmental correspondence approach to contour tone (dis) harmony patterns. In *Proceedings of the annual meetings on phonology*.

Kristina Strother-Garcia, Jerey Heinz, and Hyun Jin Hwangbo. 2017. Using model theory for grammatical inference: a case study from phonology. In *International Conference on Grammatical Inference*, pages 66–78. PMLR.

Logan Swanson, Jeffrey Heinz, and Jon Rawski. 2026. Phonotactic learning with structure, not statistics. *Linguistic Inquiry*. In press.

Mai H Vu, Ashkan Zehfroosh, Kristina Strother-Garcia, Michael Sebok, Jeffrey Heinz, and Herbert G Tanner. 2018. Statistical relational learning with unconventional string models. *Frontiers in Robotics and AI*, 5:76.

Colin Wilson and Gillian Gallagher. 2018. Accidental gaps and surface-based phonotactic learning: A case study of south bolivian quechua. *Linguistic Inquiry*, 49(3):610–623.

Moira Yip. 2002. *Tone*. Cambridge University Press.

Cheryl Zoll. 2003. Optimal tone mapping. *Linguistic Inquiry*, 34(2):225–268.

A Appendix

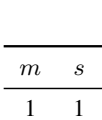
Constraint	String	Feature			AR
***RISE**$_\sigma$	LH	L:	+	−	
		C:	−	−	
		B:	0	0	
***CONTOUR**$_\mu$	R	L:	+		
		C:	+		
		B:	0		
	F	L:	−		
		C:	+		
		B:	0		
***3T**	HHH	L:	−	−	−
		C:	−	−	−
		B:	0	0	0
	LLL	L:	+	+	+
		C:	−	−	−
		B:	0	0	0
	HHL	L:	−	−	+
		C:	−	−	−
		B:	0	0	0
	HLL	L:	−	+	+
		C:	−	−	−
		B:	0	0	0
	HLH	L:	−	+	−
		C:	−	−	−
		B:	0	0	0
***NOFINHL**	HL.	L:	−	+	0
		C:	−	−	0
		B:	0	0	+
***LAPSE**	L.L	L:	+	0	+
		C:	−	0	−
		B:	0	+	0

Table A1: Baseline constraints represented in string, feature, and autosegmental representations. Note the string and featural representations for *3T would also include LHH, LHL, and LLH, but these patterns are ruled out by *RISE$_\mu$.

m	s	t	Recall	Precision	F1
1	1	1	1.000	0.500	0.667
1	1	2	1.000	0.615	0.762
1	1	3	1.000	0.615	0.762
1	2	1	1.000	0.500	0.667
1	2	2	1.000	0.615	0.762
1	2	3	1.000	0.615	0.762
1	3	1	1.000	0.500	0.667
1	3	2	1.000	0.615	0.762
1	3	3	1.000	0.615	0.762
2	1	1	1.000	0.500	0.667
2	1	2	1.000	0.727	0.842
2	1	3	1.000	0.727	0.842
2	2	1	1.000	0.500	0.667
2	2	2	1.000	0.727	0.842
2	2	3	0.969	0.721	0.826
2	3	1	1.000	0.500	0.667
2	3	2	1.000	0.727	0.842
2	3	3	0.969	0.721	0.826
3	1	1	1.000	0.615	0.762
3	1	2	1.000	1.000	1.000
3	1	3	1.000	1.000	1.000
3	2	1	1.000	0.615	0.762
3	2	2	1.000	1.000	1.000
3	2	3	0.953	1.000	0.975
3	3	1	0.984	0.612	0.754
3	3	2	0.984	1.000	0.992
3	3	3	0.938	1.000	0.967

Table A2: Performance of BUFIA-AR across complexity bounds (t, s, m).

Investigating Syntactic Biases in Multilingual Transformers with RC Attachment Ambiguities in Italian and English

Michael Kamerath and Aniello De Santo
Dept. of Linguistics
University of Utah
{michael.kamerath, aniello.desanto}@utah.edu

Abstract

This paper investigates whether monolingual and multilingual LLMs show human-like preferences when presented with examples of relative clause attachment ambiguities in Italian and English. We also test whether these preferences can be modulated by lexical factors (the type of verb/noun in the matrix clause) which have been shown to be tied to subtle constraints on syntactic and semantic relations. Our results overall showcase how LLM behavior varies inconsistently across models and languages, and highlight the importance of leveraging subtle syntactic contrasts in exploring these models' ability to correctly align with human-like preferences.

1 Introduction

A recent but already classical line of work has focused on evaluating neural models' predictions on fine-grained syntactic phenomena/constructions, in order to probe whether the models have learned knowledge about the specific structural characteristics of (a) language (Linzen et al., 2016; Marvin and Linzen, 2018; Gauthier et al., 2020; Warstadt et al., 2019, 2020b,a; Sartran et al., 2022; Newman et al., 2021a; Jumelet et al., 2024; Arora et al., 2024). This type of comparison leverages psycholinguistic design to gain insight into the opaque (learned or architectural) biases of LLM (Linzen and Baroni, 2021; Futrell et al., 2019; Ettinger, 2020).

While a majority of past work has focused on evaluating LLMs' syntactic knowledge in terms of their ability to distinguish grammatical and ungrammatical constructions, an important component of human sentence comprehension is ambiguity resolution (Altmann, 1998; Gibson and Pearlmutter, 1998). In particular, it is worth investigating how neural models handle multiple *simultaneously correct interpretations* for a single sentence in the absence of disambiguating cues/context (Davis and van Schijndel, 2020; Bhattacharya et al., 2022; Liu et al., 2023; Zhou et al., 2024).

Consider the case of the relative clause (RC) "*that was running*" following the complex noun phrase "*son of the doctor*", as in (1):

(1) I saw the son of the doctor that was running.

There are two possible interpretations of this sentence: in one interpretation the RC modifies *the doctor* (low attachment; LA), and in the other it modifies *the son* (high attachment; HA). Famously, human preferences for HA or LA vary both individually and cross-linguistically, and are affected by a variety of syntactic and semantic factors (Cuetos and Mitchell, 1988; De Vincenzi and Job, 1993; Mitchell et al., 1990; Miyamoto, 1998; Maia et al., 2007; Abdelghany and Fodor, 1999). Moreover, some of these factors (e.g., the type of verb used in the matrix clause of the sentence) seem to be tied to subtle syntactic differences in each language (Cinque, 1992; Grillo et al., 2015; Grillo and Costa, 2014, a.o.).

RC attachment ambiguities thus present an interesting way of probing LLMs' syntactic knowledge and behavior. In fact, investigating LLMs' performance over ambiguous sentences cross-linguistically might provide crucial insights into the kind of linguistic biases available to these models through their training data, and the properties of the models tied to architectural choices (Davis and van Schijndel, 2020; Li et al., 2024). As differences in the frequency of HA vs. LA structures have been argued to account for the cross-linguistic variation of RC preferences at least to some degree, it seems reasonable that LLMs would be able to replicate (some of) these patterns. However, RC attachment seems to be understudied in the LLM syntactic evaluation literature (Davis and van Schijndel, 2020; Issa and Atouf, 2024; Lee et al., 2025; Scheinberg et al., 2025).

Here, we aim to add to this scarce literature, and evaluate a variety of LLMs to determine their disambiguation strategies for RCs in Italian and English. We compare Italian to English since the two lan-

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 191–218
July 3-4, 2026 ©2026 Association for Computational Linguistics

guages have some shared structural properties (e.g., SVO, post-nominal RCs), but differ in RC interpretation: modulo other variables, English speakers generally exhibit a LA RC preference while Italian speakers a HA one (Frazier, 1983; Cuetos and Mitchell, 1988; De Vincenzi and Job, 1993). Additionally, Italian and English speakers respond differently to other variables affecting RC attachment, which have been argued to be also captured by some multilingual LLMs (e.g., type of matrix verb; Grillo et al., 2015; Grillo and Costa, 2014; Henot-Mortier, 2023). Therefore, building on the psycholinguistics and LLM literature on RC attachment, here we ask:

1. whether monolingual and multilingual LLMs tested on Italian and English show any type of attachment preference when presented with ambiguous RCs;

2. whether these preferences conform to those of Italian/English speakers;

3. whether these preferences show sensitivity to fine-grained structural information modulated by properties of the matrix clause.

2 Related Work

The cross-linguistic variability of attachment preferences for ambiguous RCs has been focus of many psycholinguistics debates, due to its direct relevance to questions about the mechanisms guiding human sentence processing (Frazier, 1983; Cuetos and Mitchell, 1988; De Vincenzi and Job, 1993; Gibson and Pearlmutter, 1998; Grillo et al., 2015; Hemforth et al., 2015; Lee and De Santo, 2024, a.o.).

Famously, when presented with a globally ambiguous sentence like in (1), and in the absence of a disambiguating context, English speakers tend to prefer a LA interpretation: an interpretation in which the RC gives us information about (*modifies)* the second noun of the preceding complex noun phrase (Frazier, 1983; Cuetos and Mitchell, 1988). This LA preference is well attested in other languages including Mandarin Chinese and Arabic (Shen, 2006; Abdelghany and Fodor, 1999; Ehrlich, 1999). In turn, a preference for the RC modifying the first noun — a HA interpretation — has been found in languages like Italian, Spanish, and Dutch (Cuetos and Mitchell, 1988; De Vincenzi and Job, 1993; Brysbaert, 1996; Frenck-Mestre and Pynte, 2000; Mitchell et al., 2000). Beyond these broader preferences at the language level, multiple factors have been shown to affect RC preferences across languages — for instance, referentiality of the modified nouns, lexical and structural frequency, and semantic or pragmatic plausibility (De Vincenzi and Job, 1993; MacDonald et al., 1994; Gilboy et al., 1995; Ferreira, 2003; Fernández, 2003; Swets et al., 2008; Acuna-Farina et al., 2009).

Recently, it has been argued that one important predictor of attachment disambiguation 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). More generally, reviewing past literature on RC attachment preferences in so-called HA languages, Grillo and Costa (2014) have related this verb-type sensitivity to a subtle structural ambiguity at the complementiser, beyond the classic LA RC vs. HA RC choice. Some languages allow for a construction known as a Pseudo-Relative Clause (PRs), which is string-identical to RCs but different at the semantic, syntactic, and prosodic levels (Cinque, 1992; Grillo and Costa, 2014; Aguilar and Grillo, 2021, a.o.). In particular, instead of providing information about the entity (noun) that is modified (2a), PRs denote direct perception of events (2b) and are thus only compatible with some specific subclasses of verbs (e.g., *photograph, record*) in the matrix clause (perception verbs, introducing events).

(2) Gianni vide il figlio che correva.

a. Gianni vide [$_{DP}$il [$_{NP}$figlio [$_{RC}$che correva]]].
Gianni saw [the [son [that ran]]].

b. Gianni vide [$_{SC}$[$_{DP}$il figlio] [che correva]].
Gianni saw [[the son] [running]].

While RCs modify nouns, PRs are analyzed as complements of perceptual verbs and are only compatible with what looks like a HA interpretation, leading to an apparent HA preference with verbs that license them (Cinque, 1992, a.o.). This hypothesis has found experimental support in a variety of languages including Italian (Grillo and Costa, 2014; Lee and De Santo, 2024) and Spanish (Aguilar and Grillo, 2021). RC attachment thus offers ways to explore the sensitivity of LLMs to important structural and semantic features within and, crucially, across languages.

Starting with Linzen et al. (2016), there has been a fruitful line of research using psycholinguistic tasks to explore neural models' knowledge of different lexical, structural, and semantic linguistic properties (Marvin and Linzen, 2018; Gauthier et al., 2020, 2022; Warstadt et al., 2019, 2020b,a; Sartran et al., 2022; Newman et al., 2021b; Jumelet et al., 2021; Arora, 2022; Gulordava et al., 2018; Sinclair et al., 2022; Goldberg, 2019; Wilson et al., 2023) — and to evaluate whether model behavior resembles the performance of humans tested on similar tasks/constructions (**?**Futrell et al., 2019; Ettinger, 2020).

While some work probing LLMs' ability to deal with (different types of) ambiguity exists (Van Schijndel and Linzen, 2018; Bhattacharya et al., 2022; Liu et al., 2023; Zhou et al., 2024; Li et al., 2024), little attention has been paid to the phenomenon of RC attachment in absence of a disambiguating context. In this sense, Davis and van Schijndel (2020) analyzed the ability of LSTMs to learn RC attachment preferences in English and Spanish. They showed that LSTMs preferred English-like attachment (LA) in both English and Spanish. More recently, Issa and Atouf (2024) tested RC attachment in Arabic with a variety of transformer models (Vaswani et al., 2017), using a zero-shot prompting method. They showed significant variability across model architectures, with some models' behavior being in line with the attachment preferences reported for Arabic speakers, while others showing no preference at all. Similarly, Lee et al. (2025) evaluated whether world knowledge biases help RC attachment disambiguation in six typologically diverse languages. Using prompting methods that simulated a force-choice decision task, they showed how three models (Claude 3.5 Sonnet, GPT-4o, and Llama 3.1) exhibited a strong attachment preference across languages (see also Scheinberg et al., 2025). Furthermore, going back to our discussion of linguistic factors that modulate RC preferences, Henot-Mortier (2023) has shown that monolingual and multilingual transformer architectures exhibit some sensitivity to PR-availability in French. However, this work evaluated PR-related properties only in contexts outside of ambiguous RC, and it thus unclear whether they would modulate an LLM's choice of attachment.

In sum, the complex interaction between RC attachment and other syntactic/semantic factors opens an exciting set of possibilities for the cross-linguistic evaluation of LLMs' linguistic performance. In what follows, building on the results of Davis and van Schijndel (2020) and Henot-Mortier (2023), we focus on evaluating a set of monolingual and multilingual models on the patterns of RC-attachment and PR-sensitivity reported in the psycholinguistic literature for Italian and English (Grillo and Costa, 2014; Grillo et al., 2015; Lee and De Santo, 2024).

Model Name	Lang.	Reference
GePpeTto	IT	De Mattei et al. (2020)
AlBERTo	IT	Polignano et al. (2019)
UmBERTo	IT	Parisi et al. (2020)
bert-base-multilingual-cased	M	Devlin et al. (2019a)
xlm-mlm-17-1280	M	CONNEAU and Lample (2019)
XLM-RoBERTa-large	M	Conneau et al. (2020)
GPT-2	ENG	Radford et al. (2019)
bert-base-uncased	ENG	Devlin et al. (2019b)

Table 1: Monolingual (IT: Italian; ENG: English) and Multilingual (M) models tested in this paper.

3 Italian Experiment

As mentioned, past literature suggests that in languages that allow for PRs (e.g., Italian) if the matrix verb is perceptual a PR parse takes precedence, resulting in a HA interpretation. Otherwise, a preference for an LA intepretation is observed. Grillo and Costa (2014) tested this prediction by evaluating Italian participants' behavior when exposed to globally ambiguous sentences containing a complex noun phrase followed by an RC. Sentences varied over the type of verb used in the matrix clause (perceptual/stative). As predicted, participants showed a HA preference only with perceptual verbs, and exhibited a LA preference with stative verbs.

Here, we exploit this design to explore whether the type of matrix verb in Italian sentences affects LLM attachment preferences. In the past, a common evaluation technique has been to check whether a model assigns a higher probability to a grammatical sentence compared to an ungrammatical one (Linzen et al., 2016; Gulordava et al., 2018). However, here we are interested in probing an LLM's preference in choosing one grammatical interpretation over another equivalently grammatical one, in the absence of other disambiguating factors (e.g., context). To do so, instead of using the globally ambiguous sentences of Grillo and Costa (2014), we follow Davis and van Schijndel (2020) and adopt sentences that are temporarily ambiguous.

Specifically, we leverage a modification of the Grillo and Costa (2014)'s stimuli presented by Lee and De Santo (2024). We follow a 2×2 design, in which quartets of sentences vary across two

dimensions: Verb Type and Attachment Type. As in Grillo and Costa (2014), sentences include a main verb which is either perceptual (*heard*) or stative (*worked with*) and a complex noun phrase (*the grandma of the girls*) followed by an RC. Items are disambiguated towards HA or LA based on singular/plural agreement between one of the nouns in the matrix clause (*grandma/girls*), and the embedded verb. This is possible since Italian differentiates singular/plural morphology explicitly on the main verb. We use Lee and De Santo (2024)'s items, which include 24 sentence sets for a total of ninety-six sentences. Each set contains 4 sentences varying across the two dimensions mentioned above (Appendix A). In line with the models tested for French by Henot-Mortier (2023), we test three Italian-only models, and three multilingual models (GePpeTto; AlBERTo; bert-base-multilingual-cased; xlm-mlm-17-1280; XLM-roBERTa-large; see Table 1).

3.1 Surprisal Analysis

Following Davis and van Schijndel (2020), we evaluate LLMs using information-theoretic surprisal (Hale, 2001; Levy, 2008), which is usually defined as the inverse log probability assigned to a word in a sentence given its preceding context. Fixed verb-type, our stimuli include sentence pairs that are string identical until the singular/plural features on the disambiguating verb. Thus, for each item in the dataset, we compute surprisal at the embedded verb using the `minicons` library (Misra, 2022).[1] For the xlm-mlm–17-1280 model, three of the sets were excluded due to `minicons` not returning a surprisal value for the disambiguating verb.

For each LLM, we fit a linear mixed-effect (LME) model using Surprisal at the embedded verb as the dependent variable, and Verb Type and Attachment Type as fixed effects. We also include a random intercept for set, in order to account for lexical variation across sentence quartets.[2]. All analyses were performed using the lme4 package (version 1.1.35.5; Bates et al., 2015) in R (R version 4.4.1; R Core Team, 2021).

Statistical analyses show no significant attachment or verb type effects, nor their interaction, for any of the LLMs tested (see Figure 1).[3] These results can be interpreted as the absence of an attachment preference (in line with Italian speakers or not), and a lack of sensitivity to verb-type properties, in both the Italian-only and the multilingual models.

4 English Experiments

Since PR availability co-varies in Italian with semantic properties of the matrix verb (i.e., implicit causality), Grillo et al. (2015) acknowledge that a pragmatic explanation for their results could also be viable (Gilboy et al., 1995; Rohde et al., 2011). They conducted an English study probing similar variables modulating RC attachment as those manipulated in the Italian studies discussed above.

While not a PR-language, English allows for structures that are interpretatively similar to PRs — eventive small clauses (SC), which describe dynamic events as in: *Star heard the monster scream*. These constructions are licensed by perceptual verbs, and have many of the functional and interpretative properties of PRs in Italian. However, English SCs are not string equivalent to RC clauses with an explicit complementiser. Therefore, a PR-based account of attachment preferences predicts that PR-related verb-type effects should not arise with RC sentences in English. On the other hand, a pragmatic account would predict a perceptual/stative effects on attachment in English as observed by Grillo and Costa (2014) in Italian.

In an offline questionnaire, Grillo et al. (2015) showed that English participants consistently prefer LA, although they observed a small HA boost in the SC-licensing/Perceptual verb condition. They argue that these results are incompatible with a pragmatic account of the Italian findings, strengthening the PR-availability argument for their Italian results. This design offers us a way to further probe factors affecting RC attachment strategies in LLMs with a direct cross-linguistic comparison of the manipulated variables. Additionally, as implicit causality has been explored in the LLM literature to somewhat conflicting results, this experimental set up might lead to broader insights into LLMs' sensitivity to semantic/pragmatic variables (Kankowski et al., 2025; Ke-

[1] Specifically, for each of the LLMs tested, verbs were tokenized into stems and suffixes (e.g. *gridava/vano* is tokenized as [*grida*] and [*va/vano*]). Each LLM was verified to ensure tokenization occurred in this form. All LLMs tested were consistent with this tokenization approach with the exception of the *xlm-mlm-17-1280* model, which tokenized some verbs to include part of the stem in the token. For example, *saltava* was tokenized as [*sal*] and [*tava*] rather than [*salta*] and [*va*]. This linguistically opaque tokenization occurred for 4 out of the 48 Italian stimuli of the form *verb+va*.

[2] `Surprisal ~ Verb Type + Attachment Type + Verb Type*Attachment Type + (1|set)`

[3] For the sake of space, when discussing analyses we only report significance. The full output of each LME model can be found in Appendix B. Scripts and data for all the experiments in this paper can be found at https://shorturl.at/yYgIf

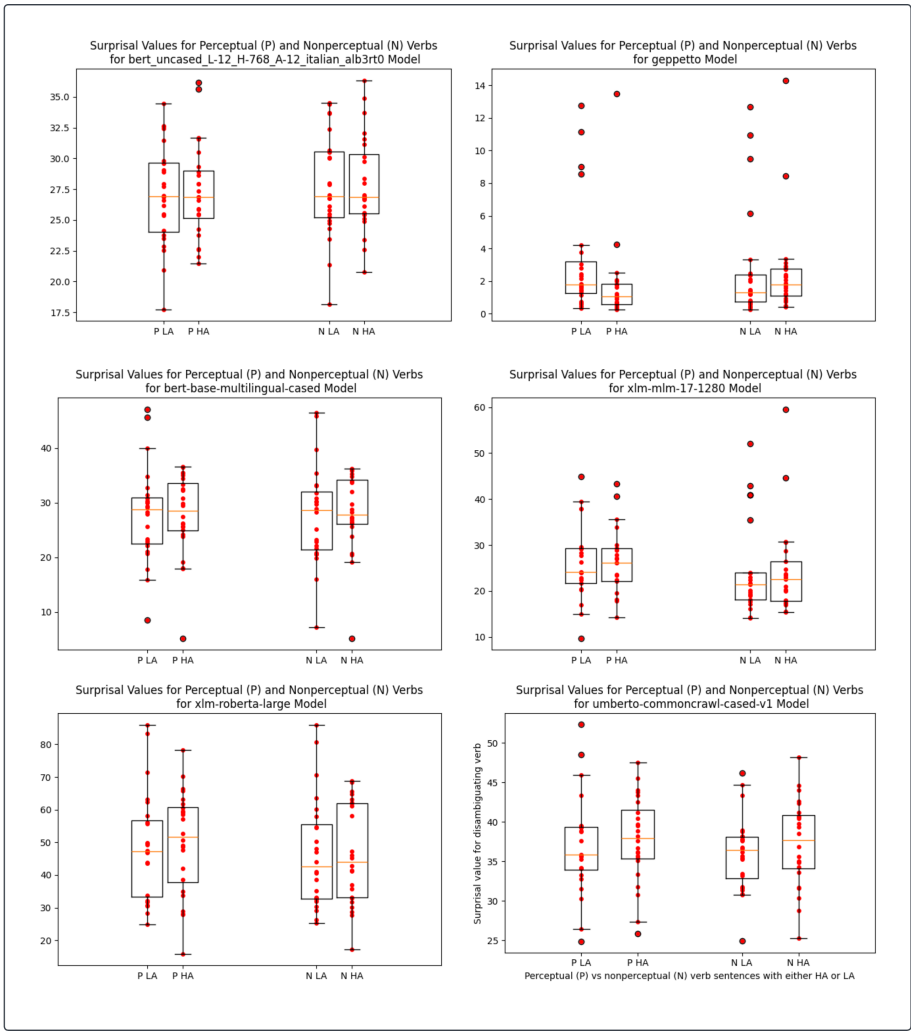


Figure 1: Surprisal values by condition, for each one of the models tested in the Italian Experiment.

mentchedjhieva et al., 2021). In addition to the verb-type manipulation of the Italian experiments, Grillo et al. (2015) also modulate the type of nominal used as the first noun in the complex noun — either licensing a SC or not (*heard* vs. *scream*). This noun-type manipulation implies testing RCs following a complex noun-phrase in the subject position of the main sentence, compared to the object modifying RCs used when manipulating verb type (Example 3).

(3) a. Kelly heard the grandma of the girl that was screaming.

b. The sounds of the grandma of the girl that was screaming is annoying.

We adopt this design, but we will again depart from the baseline psycholinguistic study by using disambiguated RCs over globally ambiguous ones. We thus split Grillo et al. (2015)'s experiment into two: Experiment 1 will test the effect of verb-type in English, while Experiment 2 will text the effects of noun-type/RC position. In line with the Italian experiment, we test two monolingual English models, and the three multilingual models evaluated in Section 1.

4.1 Experiment 1: Verb-Type Effects

First, we investigate Verb Type effects in English, using stimuli adapted from the first experiment in Grillo et al. (2015). These include sets of four lexically matched items holding all properties of a sentence constant except for the matrix verb, which is either a RC-only verb or a SC-licensing verb (Appendix A). Grillo et al. (2015) report that human participants tested on these stimuli showcase a general preference for LA, but a slight HA boost in the SC-licensing condition.

Similarly to the Italian experiment, all items were modified to disambiguate LA/HA based on singular/plural agreement on the embedded verb. Because of the properties of English, this disambiguation happens over an auxiliary verb

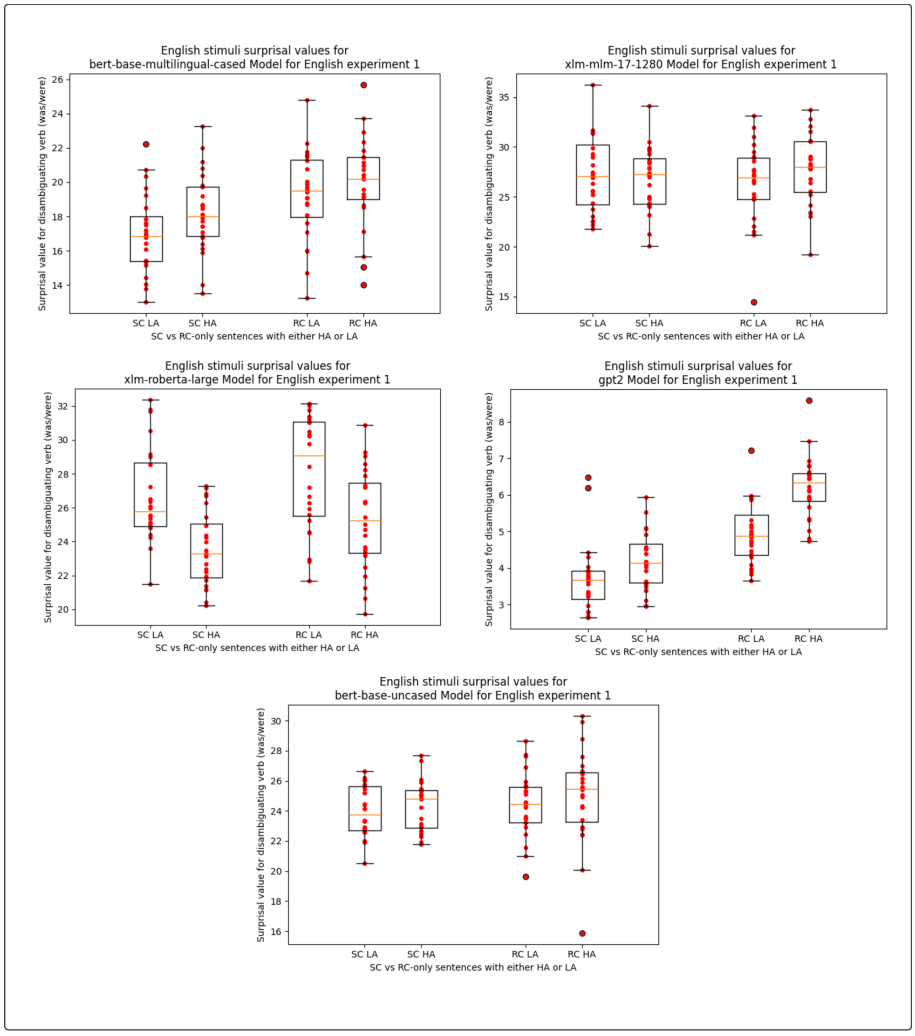


Figure 2: Surprisal values by condition, for each one of the models tested for Verb-type Effects in English (Exp. 1).

(*was/were*) instead of directly on the embedded verb. Experimental stimuli included twenty-four sets, for a total of ninety-six sentences. We fit an identical linear mixed-effect model model as for the Italian Experiment, using Surprisal at the embedded verb as the dependent variable, and Verb Type and Attachment Type as fixed effects.[4]

Compared to the Italian Experiments, results here are more mixed (Figure 2). For the multilingual bert-base model, we found a significant Verb Type effect, consistent with surprisal values being generally lower in the SC-licensing verb condition that in the RC-only condition. These differences are independent of Attachment Type, although with SC verbs we observe a (non-significant) trend in favor of the LA condition — which is line with the known LA preference in English, but somewhat in contrast with what Grillo et al. (2015) found with human participants. No significant effects were found with the xlm model, but there were marginal effects of Attachment Type and of the Verb Type/Attachment Type interaction. The xlm model's results do trend towards lower surprisal for LA in the RC-only condition (Figure 2). For the RoBERTa model we found significant Very Type and Attachment Type effects, but no interaction effects. Again, surprisal values in the SC condition are lower independently of Attachment Type (Figure 2). Additionally, surprisal values for HA items are significantly lower than those of LA items (thus indicating a HA preference). Considering now the monolingual models, with GPT-2 we found both Verb-Type and Attachment-Type effects. These results are related to higher surprisal values in the RC-Only and the HA conditions. We additionally found a significant interaction effect. Descriptively this is the closest the pattern of results in human experiments. It is worth observing the magnitude of these effects: human participants

[4] `Surprisal ~ Verb Type + Attachment Type + Verb Type*Attachment Type + (1|set)`

still exhibit a certain rate of HA choices in both conditions, compared to the heavy LA tendency of GPT-2. Finally we found no significant effects for bert-base-uncased across conditions.

4.2 English Experiment 2: Noun-Type Effects

In a second experiment, we leverage the stimuli in the nominal condition of Grillo et al. (2015)'s first experiment. This condition compares nominals that license SC (i.e., event-takin nouns compatible with the description of an event, like "*picture*" or "*sound*") to nominals that are only compatible with RCs (e.g., "*car*" , "*comb*"). As mentioned above, the nominal condition is also designed so that the complex noun phrase (and thus the following RC) occupies the subject position of the matrix clause (as in Table 4). On these stimuli, English participants show a LA preference, but no noun-type effect (Grillo et al., 2015).

For our LLM tests, we again modify all items to disambiguate LA/HA based on singular/plural agreement on the embedded verb, resulting in a 2×2 design. Results from linear-mixed effect models[5] for each LLM are mixed, but generally in line with those in the first English experiment (Figure 3). For the multilingual bert model, we find a strong effect of Attachment Type, no effect of Noun Type, and no interaction. These are compatible with bert strongly preferring LA items independently of the noun manipulation. For the xlm model, we found a significant clause type effect, but no effect of attachment, nor an interaction. This is compatible with a tendency for slightly higher surprisal values with SC-licensing nominals independently of Attachment Type. For the RoBERTa model we again found a strong Attachment Type effect, with no interaction with Noun Type. This is the result of a strong preference for HA items across Noun-Type conditions (Figure 3), in line with Roberta's behavior in Experiment 1. As for the English-only models, with GPT-2 we found a significant effect of Attachment type. Similarly, bert-base-uncased presents a significant sensitivity to attachment type and no other effect. It is also interesting to observe that although monolingual and multilingual bert behave similarly in their trends, their patterns are not human like.

[5] `Surprisal ~ Noun Type + Attachment Type + Noun Type*Attachment Type + (1|set)`

5 Discussion and Further Work

This paper contributes to work evaluating the sensitivity of LLMs to a variety of linguistic properties, by exploring whether/how the preference for High or Low attachment of RCs is affected by syntactic/semantic properties of the matrix clause. Our results overall showcase how LLM behavior does not align with human data and, more importantly, varies inconsistently across models and languages.

Specifically, we measured the difference in surprisal of locally ambiguous sentences at the point of disambiguation (the embedded verb) to determine whether a number of (monolingual and multilingual) LLMs align with human-like RC attachment preferences in Italian and English. Furthermore, we tested whether these preferences can be modulated by lexical factors in the matrix clause (Verb Type or Noun Type), which have been argued to be related to subtle differences between RCs and other constructions.

For Italian, statistical results over raw surprisal values indicate that none of the models we tested exhibits any attachment preference at all, whether in line with the human results or not. Furthermore, we do observe high item-level variability, which is also not human-like. Even though we control for item-level lexical effects in our statistical models, because of this stark item-based variability we do note interesting (non statistically significant) tendencies in some of the models that beg for deeper inquiry in future work. For instance, modulo some high surprisal LA items, the GePpeTto model shows a general qualitative preference towards LA, in particular with perceptual verbs.

English results across two experiments were more mixed. While some models showcase some type of attachment preference, and at times verb and noun type effects on these preferences, these were not exactly in line with human data. For instance, while the multilingual bert-base model does show a slight preference for LA items, the RoBERTa model shows a strong bias towards HA items, in contrast with the reported LA preference for English. The mirrored behavior of BERT and RoBERTa across the two English experiments is also of note, and opens question for future comparisons — as does the fact that surprisal values across models were slightly higher in the RC-only condition.

Notably, our statistical results are also somewhat in contrast with what previous work found for Spanish and Arabic (Davis and van Schijndel, 2020; Issa and Atouf, 2024), which found a general LA

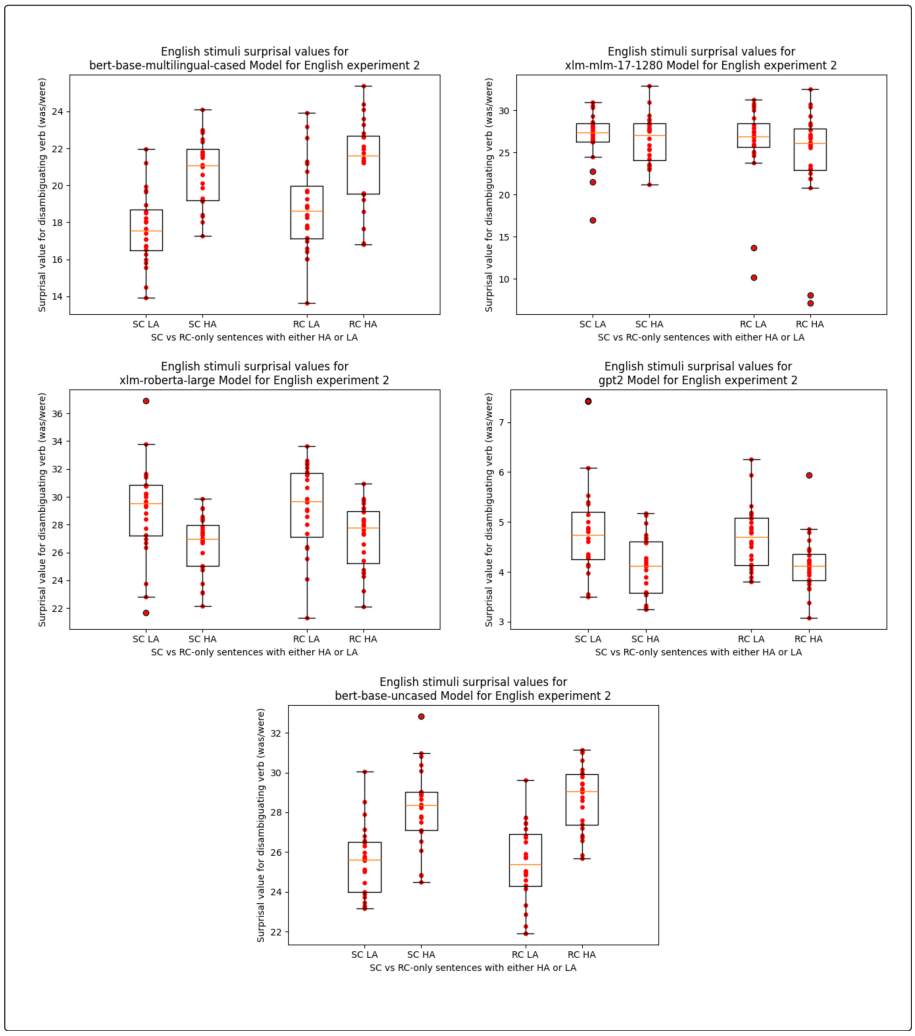


Figure 3: Surprisal values by condition, for each one of the models tested for Noun-Type effects in English (Exp. 2).

preference even in HA languages. 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 then probe differences between architectures and tasks/measures more in depth. With respect to our comparison with some of the recent literature on LLM attachment preferences, it is worth considering the effects of task. Some studies on LLMs and RCs have adopted prompting (Brown et al., 2020) as a method of studying LLMs's sensitivity to RC properties in ambiguous attachment contexts (Issa and Atouf, 2024; Scheinberg et al., 2025; Lee et al., 2025). Importantly, prompting tasks have been criticized with respect of their ability to provide insights into the linguistic representations and decision processes of LLMs (Hu and Levy, 2023; Hu and Frank, 2024). Crucially, prompting tasks rely on specific meta-linguistic information that might confound the evaluation of model sensitivity to linguistic properties. For instance, Lee et al. (2025) set-up an evaluation pipe-line dependent on a model's ability to first identify an RC and DPs related to it. Generally, this tension between past work and our results suggest the necessity of testing similar constructions while keeping architectural (and task) details constant.

Overall, our work suggests a primary role for structural ambiguity in the study of LLMs' capabilities cross-linguistically, problematizing general, broad-stroke claims about "human-like" performance of LLMs, and strengthening the argument in favor of psycholinguistically motivated, cross-linguistics benchmarks. We argue for the importance of studying LLM performance over ambiguous, grammatically well-formed linguistic stimuli, as a fine-grained lens into LLM knowledge and processes.

Limitations

In this paper we argued for the importance of leveraging well-designed psycholinguistic stimuli in investigating LLM's behavior. Therefore, we relied on experimental items available from two psycholinguistic studies of interest in conducting our evaluations. However, this meant that the number of items used in the paper is relatively low compared to the size of test sets in the LLM literature (but comparable to other work on LLM/RC attachment). This choice might also have made our results particularly sensitive to item-level (set) variability, which deserves further investigation. Additionally, a limitation of comparing Italian to English is that in Italian surprisal is measured at the disambiguating verb, which varies lexically across sets, but in English the disambiguating continuation is always measured on the *was/were* contrast consistently. While we controlled for set in our statistical models to partially address item-level confounds, future work on RC attachment and noun/verb type effects should explore this difference, and be extended to multiple languages with and without pseudo-relative constructions.

Acknowledgments

We thank members of the Utah Computational Linguistics Group, the audiences at the Utah Undergraduate Linguistics Conference 2025 and at the University of Utah's first "SPEAKING TO MACHINES: Cross- Disciplinary dialogues in Linguistics, AI, and Computer Science" workshop, and SCiL's anonymous reviewers for helpful feedback on this project.

References

Hala Abdelghany and Janet Dean Fodor. 1999. Low attachment of relative clauses in arabic. *Poster presented at AmlaP (Architectures and mechanisms of language Processing), edinburgh, uk*.

Carlos Acuna-Farina, Isabel Fraga, Javier García-Orza, and Ana Piñeiro. 2009. Animacy in the adjunction of spanish rcs to complex nps. *European Journal of Cognitive Psychology*, 21(8):1137–1165.

Miriam Aguilar and Nino Grillo. 2021. Spanish is not different: On the universality of minimal structure and locality principles. *Glossa: a journal of general linguistics*, 6.

Gerry TM Altmann. 1998. Ambiguity in sentence processing. *Trends in cognitive sciences*, 2(4):146–152.

Aryaman Arora. 2022. Universal Dependencies for Punjabi. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5705–5711, Marseille, France. European Language Resources Association.

Aryaman Arora, Dan Jurafsky, and Christopher Potts. 2024. CausalGym: Benchmarking causal interpretability methods on linguistic tasks. pages 14638–14663.

Douglas Bates, Martin Maechler, Ben Bolker, Steven Walker, Rune Haubo Bojesen Christensen, Henrik Singmann, Bin Dai, Gabor Grothendieck, Peter Green, and M Ben Bolker. 2015. Package 'lme4'. *convergence*, 12(1):2.

Sunit Bhattacharya, Vilém Zouhar, and Ondrej Bojar. 2022. Sentence ambiguity, grammaticality and complexity probes. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 40–50, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Marc Brysbaert. 1996. Modifier attachment in sentence parsing: Evidence from dutch. *The Quarterly Journal of Experimental Psychology: Section A*, 49(3):664–695.

Guglielmo Cinque. 1992. *The pseudo-relative and ACC-ing constructions after verbs of perception*. Università degli studi di Venezia.

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.

Alexis CONNEAU and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Fernando Cuetos and Don C. Mitchell. 1988. Cross-linguistic differences in parsing: Restrictions on the use of the late closure strategy in spanish. *Cognition*, 30(1):73–105.

Forrest Davis and Marten van Schijndel. 2020. Recurrent neural network language models always learn

English-like relative clause attachment. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1979–1990, Online. Association for Computational Linguistics.

Lorenzo De Mattei, Michele Cafagna, Felice Dell'Orletta, Malvina Nissim, and Marco Guerini. 2020. Geppetto carves italian into a language model. In *Proceedings of the Seventh Italian Conference on Computational Linguistics (CLiC-it 2020)*, pages 97–104.

Marica De Vincenzi and Remo Job. 1993. Some observations on the universality of the late-closure strategy. *Journal of Psycholinguistic Research*, 22:189–206.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. 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.

Karen Ehrlich. 1999. Low attachment of relative clauses: New data from swedish, norwegian and romanian. In *the 12th Annual CUNY Conference on Human Sentence Processing. New York, NY., 1999*.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Eva M Fernández. 2003. Bilingual sentence processing.

Fernanda Ferreira. 2003. The misinterpretation of noncanonical sentences. *Cognitive Psychology*, 47(2):164–203.

Lyn Frazier. 1983. Processing sentence structure. *Eye movements in reading: Perceptual and language processes*, pages 215–236.

Cheryl Frenck-Mestre and Joël Pynte. 2000. Chapter 21 - 'romancing' syntactic ambiguity: Why the french and the italians don't see eye to eye. In Alan Kennedy, Ralph Radach, Dieter Heller, and Joël Pynte, editors, *Reading as a Perceptual Process*, pages 549–564. North-Holland, Oxford.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Elodie Gauthier, Papa Séga Wade, Thierry Moudenc, Patrice Collen, Emilie De Neef, Oumar Ba, Ndeye Khoyane Cama, Ahmadou Bamba Kebe, Ndeye Aissatou Gningue, and Thomas Mendo'O Aristide. 2022. Preuve de concept d'un bot vocal dialoguant en wolof (proof-of-concept of a voicebot speaking Wolof). In *Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*, pages 403–412, Avignon, France. ATALA.

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.

Edward Gibson and Neal J Pearlmutter. 1998. Constraints on sentence comprehension. *Trends in cognitive sciences*, 2(7):262–268.

Elizabeth Gilboy, Josep-MMaria Sopena, Charles Cliftrn, and Lyn Frazier. 1995. Argument structure and association preferences in spanish and english complex nps. *Cognition*, 54(2):131–167.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Nino Grillo and João Costa. 2014. A novel argument for the universality of parsing principles. *Cognition*, 133(1):156–187.

Nino Grillo, João Costa, Bruno Fernandes, and Andrea Santi. 2015. Highs and lows in english attachment. *Cognition*, 144:116–122.

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.

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*.

Barbara Hemforth, Susana Fernandez, Charles Clifton, Lyn Frazier, Lars Konieczny, and Michael Walter. 2015. Relative clause attachment in german, english, spanish and french: Effects of position and length. *Lingua*, 166:43–64.

Adele Henot-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, Gothenburg, Sweden. Association for Computational Linguistics.

Jennifer Hu and Michael Frank. 2024. Auxiliary task demands mask the capabilities of smaller language models. In *First Conference on Language Modeling*.

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*, pages 5040–5060, Singapore. Association for Computational Linguistics.

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.

Jaap Jumelet, Lisa Bylinina, Willem Zuidema, and Jakub Szymanik. 2024. Black big boxes: Do language models hide a theory of adjective order? *arXiv preprint arXiv:2407.02136*.

Jaap Jumelet, Milica Denic, Jakub Szymanik, Dieuwke Hupkes, and Shane Steinert-Threlkeld. 2021. Language models use monotonicity to assess NPI licensing. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4958–4969, Online. Association for Computational Linguistics.

Florian Kankowski, Torgrim Solstad, Sina Zarriess, and Oliver Bott. 2025. Implicit causality-biases in humans and llms as a tool for benchmarking llm discourse capabilities. *arXiv preprint arXiv:2501.12980*.

Yova Kementchedjhieva, Mark Anderson, and Anders Søgaard. 2021. John praised mary because _he_? implicit causality bias and its interaction with explicit cues in lms. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4859–4871.

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.

So Young Lee, Russell Scheinberg, Amber Shore, and Ameeta Agrawal. 2025. Who relies more on world knowledge and bias for syntactic ambiguity resolution: Humans or LLMs? 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 3484–3498, Albuquerque, New Mexico. Association for Computational Linguistics.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Andrew Li, Xianle Feng, Siddhant Narang, Austin Peng, Tianle Cai, Raj Sanjay Shah, and Sashank Varma. 2024. Incremental comprehension of garden-path sentences by large language models: Semantic interpretation, syntactic re-analysis, and attention. *Preprint*, arXiv:2405.16042.

Tal Linzen and Marco Baroni. 2021. Syntactic structure from deep learning. *Annual Review of Linguistics*, 7(1):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.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah Smith, and Yejin Choi. 2023. We're afraid language models aren't modeling ambiguity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 790–807, Singapore. Association for Computational Linguistics.

Maryellen C MacDonald, Neal J Pearlmutter, and Mark S Seidenberg. 1994. The lexical nature of syntactic ambiguity resolution. *Psychological review*, 101(4):676.

Marcus Maia, Eva M Fernández, Armanda Costa, and Maria do Carmo Lourenço-Gomes. 2007. Early and late preferences in relative clause attachment in portuguese and spanish. *Journal of Portuguese Linguistics*, 6(1).

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.

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

Don C. Mitchell, Marc Brysbaert, Stefan Grondelaers, and Piet Swanepoel. 2000. Chapter 18 - modifier attachment in dutchdutch: Testing aspects of construal theory. In Alan Kennedy, Ralph Radach, Dieter Heller, and Joël Pynte, editors, *Reading as a Perceptual Process*, pages 493–516. North-Holland, Oxford.

Don C Mitchell, Fernando Cuetos, and Daniel Zagar. 1990. Reading in different languages: Is there a universal mechanism for parsing sentences? In *Comprehension processes in reading*, pages 285–302. Routledge.

Edson Tadashi Miyamoto. 1998. *Relative clause processing in Brazilian Portuguese and Japanese*. Ph.D. thesis, Massachusetts Institute of Technology.

Benjamin Newman, Kai-Siang Ang, Julia Gong, and John Hewitt. 2021a. Refining targeted syntactic evaluation of language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3710–3723, Online. Association for Computational Linguistics.

Benjamin Newman, Kai-Siang Ang, Julia Gong, and John Hewitt. 2021b. Refining targeted syntactic evaluation of language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3710–3723, Online. Association for Computational Linguistics.

Loreto Parisi, Simone Francia, and Paolo Magnani. 2020. Umberto: an italian language model trained with whole word masking. https://github.com/musixmatchresearch/umberto.

Marco Polignano, Pierpaolo Basile, Marco De Gemmis, Giovanni Semeraro, Valerio Basile, and 1 others. 2019. Alberto: Italian bert language understanding model for nlp challenging tasks based on tweets. In *CEUR workshop proceedings*, volume 2481, pages 1–6. CEUR.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

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

H. Rohde, R. Levy, and A. Kehler. 2011. Anticipating explanations in relative clause processing. *Cognition*, 118(3):339–358.

Laurent Sartran, Samuel Barrett, Adhiguna Kuncoro, Miloš Stanojević, Phil Blunsom, and Chris Dyer. 2022. Transformer grammars: Augmenting transformer language models with syntactic inductive biases at scale. *Transactions of the Association for Computational Linguistics*, 10:1423–1439.

Russell Scheinberg, So Young Lee, and Ameeta Agrawal. 2025. Missing the cues: Llms' insensitivity to semantic biases in relative clause attachment. *Proceedings of the Linguistic Society of America*, 10(1):5902–5902.

Xingjia Shen. 2006. *Late assignment of syntax theory: Evidence from Chinese and English*. University of Exeter (United Kingdom).

Arabella Sinclair, Jaap Jumelet, Willem Zuidema, and Raquel Fernández. 2022. Structural persistence in language models: Priming as a window into abstract language representations. *Transactions of the Association for Computational Linguistics*, 10:1031–1050.

Benjamin Swets, Timothy Desmet, Charles Clifton, and Fernanda Ferreira. 2008. Underspecification of syntactic ambiguities: Evidence from self-paced reading. *Memory & Cognition*, 36:201–216.

Marten Van Schijndel and Tal Linzen. 2018. Modeling garden path effects without explicit hierarchical syntax. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, volume 40.

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

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2020a. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020b. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.

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.

Lingling Zhou, Suzan Verberne, and Gijs Wijnholds. 2024. Tree transformer's disambiguation ability of prepositional phrase attachment and garden path effects. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12291–12301, Bangkok, Thailand. Association for Computational Linguistics.

A Templates for Experimental Stimuli

Table 2: Italian stimuli by condition, adapted from Lee and De Santo (2024).

	Verb Type	Attachment	Target					
a.	Perceptual (P)	LA	*Gianni vide il figlio dei*	*medici*	*che*	*correvano*	*la*	*maratona*
			Gianni saw the son-SG of the	*doctors-PL*	*who*	*were running-PL*	*the*	*marathon*
b.	Perceptual (P)	HA	*Gianni vide il figlio dei*	*medici*	*che*	*correva*	*la*	*maratona*
			Gianni saw the son-SG of the	*doctors-PL*	*who*	*was running-SG*	*the*	*marathon*
c.	Non-Perceptual (N)	LA	*Gianni amò il figlio dei*	*medici*	*che*	*correvano*	*la*	*maratona*
			Gianni loved the son-SG of the	*doctors-PL*	*who*	*were running-PL*	*the*	*marathon*
d.	Non-Perceptual (N)	HA	*Gianni amò il figlio dei*	*medici*	*che*	*correva*	*la*	*maratona*
			Gianni loved the son-SG of the	*doctors-PL*	*who*	*was running-SG*	*the*	*marathon*

Table 3: Stimuli by condition for English Exp. 1, adapted from Grillo et al. (2015).

	Verb Type	Attachment		Target	
a.	RC-only	HA	Jim saw the son of the doctors that	was	having dinner.
b.	RC-only	LA	Jim saw the son of the doctors that	were	having dinner.
c.	SC	HA	Jim shares the house with the son of the doctors that	was	having dinner.
d.	SC	LA	Jim shares the house with the son of the doctors that	were	having dinner.

Table 4: Stimuli by condition for English Exp. 2, adapted from Grillo et al. (2015).

	Noun Type	Attachment		Target	
a.	RC-only	HA	The picture of the son of the doctors that	was	having dinner is old.
b.	RC-only	LA	The picture of the son of the doctors that	were	having dinner is old.
c.	SC	HA	The car of the son of the doctors that	was	having dinner is old.
d.	SC	LA	The car of the son of the doctors that	were	having dinner is old.

B Summary of LME Models

	Estimate	Std. Error	df	t-value	Pr(>|t|)
(Intercept)	27.94089	0.82105	31.02296	34.031	<2e-16
Verb Type	-0.63548	0.50688	69.00000	-1.254	0.214
Attachment Type	-0.19745	0.50688	69.00000	-0.390	0.698
Verb Type : Attachment Type	-1.34373	4.39996	69.00000	-0.305	0.761

(a) AlBERTo

	Estimate	Std. Error	df	t-value	Pr(>|t|)
(Intercept)	2.5262	0.6478	80.3745	3.900	0.000199
Verb Type	-0.7534	0.8093	69.0000	-0.931	0.355153
Attachment Type	0.1703	0.8093	69.0000	0.210	0.833988
Verb Type : Attachment Type	1.2688	1.1446	69.0000	1.109	0.271492

(b) GePpeTto

	Estimate	Std. Error	df	t-value	Pr(>|t|)
(Intercept)	28.1589	1.6636	32.7576	16.927	<2e-16
Verb Type	-0.4183	1.1124	69.0000	-0.376	0.708
Attachment Type	-0.5246	1.1124	69.0000	-0.472	0.639
Verb Type : Attachment Type	0.6105	1.5732	69.0000	0.388	0.699

(c) bert_base_multilingual_case

	Estimate	Std. Error	df	t-value	Pr(>|t|)
(Intercept)	24.6305	2.0505	54.8460	12.012	<2e-16
Verb Type	1.7987	2.2630	60.0000	0.795	0.430
Attachment Type	0.3184	2.2630	60.0000	0.141	0.889
Verb Type : Attachment Type	-0.4746	3.2003	60.0000	-0.148	0.883

(d) xlm-mlm-17-1280

	Estimate	Std. Error	df	t-value	Pr(>|t|)
(Intercept)	46.35441	3.29053	47.98074	14.087	<2e-16
Verb Type	3.62229	3.11124	69.00000	1.164	0.248
Attachment Type	0.08279	3.11124	69.00000	0.027	0.979
Verb Type : Attachment Type	-1.34373	4.39996	69.00000	-0.305	0.761

(e) XLM-roBERTa-large

	Estimate	Std. Error	df	t-value	Pr(>|t|)
(Intercept)	2.5262	0.6478	80.3745	3.900	0.000199
Verb Type	-0.7534	0.8093	69.0000	-0.931	0.355153
Attachment Type	0.1703	0.8093	69.0000	0.210	0.833988
Verb Type : Attachment Type	1.2688	1.1446	69.0000	1.109	0.271492

(f) UmBERTo

Table 5: LMER Summary for all models in the Italian Experiment. Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05.

	Estimate	Std. Error	df	t-value	Pr(>|t|)
(Intercept)	19.9798	0.5074	55.6898	39.377	<2e-16
Verb Type	-1.7528	0.5243	69.0000	-3.343	0.00134**
Attachment Type	-0.7160	0.5243	69.0000	-1.366	0.17648
Verb Type : Attachment Type	-0.5337	0.7414	69.0000	-0.720	0.47407

(a) bert_base_multilingual_case

	Estimate	Std. Error	df	t-value	Pr(>|t|)
(Intercept)	27.7412	0.7469	45.2554	37.144	<2e-16
Verb Type	-0.9368	0.6790	69.0000	-1.380	0.1721
Attachment Type	-1.3458	0.6790	69.0000	-1.982	0.0515
Verb Type : Attachment Type	1.8169	0.9602	69.0000	1.892	0.0627

(b) xlm-mlm-17-1280

	Estimate	Std. Error	df	t-value	Pr(>|t|)
(Intercept)	25.2791	0.5831	53.7553	43.350	<2e-16
Verb Type	-1.7132	0.5907	69.0000	-2.900	0.00499**
Attachment Type	2.8251	0.5907	69.0000	4.783	9.46e-06***
Verb Type : Attachment Type	0.2826	0.8353	69.0000	0.338	0.73616

(c) xlm-roberta-large

Table 6: LMER Summary for all models in the English Experiment 1. Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05.

	Estimate	Std. Error	df	t-value	Pr(>|t|)
(Intercept)	21.2704	0.4423	47.9400	48.094	<2e-16
Noun Type	-0.5328	0.4179	69.0000	-1.275	0.207
Attachment Type	-2.4803	0.4179	69.0000	-5.935	1.06e-07***
Noun Type : Attachment Type	-0.5808	0.5911	69.0000	-0.983	0.329

(a) bert_base_multilingual_case

	Estimate	Std. Error	df	t-value	Pr(>|t|)
(Intercept)	24.6088	0.9001	51.2856	27.339	<2e-16
Noun Type	1.9332	0.8871	69.0000	2.179	0.0327*
Attachment Type	1.5127	0.8871	69.0000	1.705	0.0926
Noun Type : Attachment Type	-1.1707	1.2545	69.0000	-0.933	0.3540

(b) xlm-mlm-17-1280

	Estimate	Std. Error	df	t-value	Pr(>|t|)
(Intercept)	27.1706	0.5632	44.8643	48.243	<2e-16
Noun Type	-0.5667	0.5089	69.0000	-1.114	0.269274
Attachment Type	2.0496	0.5089	69.0000	4.028	0.000143***
Noun Type : Attachment Type	0.3905	0.7197	69.0000	0.589182	0.589182

(c) xlm-roberta-large

Table 7: LMER Summary for all models in the English Experiment 2. Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05

C Additional Analyses

C.1 Preferences as Categorical Pairwise Comparisons

Following standard practices in psycholinguistics, for the paper's core analyses statistical robustness of the effects/contrasts has been determined by running linear-mixed effects models using the original distribution of surprisal values over items. This is also consistent with what is done with human data during (for instance) online tasks involving locally ambiguous sentences like the ones we used, and it is also how Davis and van Schijndel (2020) discuss significant effects for their results. However, a qualitative understanding of model's trend, in line with results from human participants from forced choices tasks targeting globally ambiguous sentences, can be achieved by coding a model's preference for HA/LA categorically for each item pair in a set (Davis and van Schijndel, 2020). That is, in each set items can be paired by keeping Verb Type/Noun Type consistent. Then, if surprisal for the LA disambiguated item was lower than the surprisal for the HA disambiguated item, attachment is coded as LA. See Example 4 for a summary of this coding approach across the experiments in this paper.

(4) Interpretation of Pairwise comparisons for each experiment

a. `Attachment Preference` ← `LOW` if Verb Surprisal(a) > Verb Surprisal(b)

b. `Attachment Preference` ← `HIGH` if Verb Surprisal(a) < Verb Surprisal(b)

c. `Attachment Preference` ← `LOW` if Verb Surprisal(c) > Verb Surprisal(d)

d. `Attachment Preference` ← `HIGH` if Verb Surprisal(c) < Verb Surprisal(d)

C.1.1 Italian Pairwise Comparisons

In terms of qualitative interpretation, when comparing sentence types a higher percentage of low surprisal values for LA items compared to their paired HA items would indicate a LA preference, and vice versa. Verb-type sensitivity would show these high/low surprisal values at the embedded verb also modulated by the properties of the matrix verb. Figure 4 shows a visualization of model preferences given this kind of coding schema in the Italian Experiment. In contrast with the quantitative analyses over raw surprisal scores, these qualitative contrasts do show some trends compatible with attachment preferences — although still not exactly aligned with human data. In particular, we observe a profile compatible with a LA preference for the Perceptual condition for GePpeTto and AlBERTo. The xlm and RoBerta models show an opposite trend, with an LA-leaning pattern in the Non-Perceptual condition. Finally, the Umberto and multilingual bert-base models show an HA profile across both Verb-Type conditions.

C.1.2 English Pairwise Comparisons: Exp. 1

For the first English experiment, we similarly observe differences between the quantitative comparisons reported in the main paper and the qualitative pairwise differences. Notably, it seems that the roberta model prefers HA items in almost every set — in contrast with the pattern of preferences usually reported for human English participants (see Figure 5). This is showcased in the model complete lack of HA choices in the RC-Only condition, and in a small increase in HA rates in the SC condition.

It is also interesting to observe that the magnitude of these effects is strikingly different: human participants still exhibit a certain rate of HA choices in both conditions, compared to the heavy LA tendency of GPT-2. Finally, we can observe a LA trend in the qualitative comparisons for bert-base-uncased across conditions, in contrast with the not significant statistical effects.

C.1.3 English Pairwise Comparisons: Exp. 2

For the second English Experiment, qualitative contrasts for multilingual bert showcase an almost absolute preference for LA attachment across all conditions (see Figure 6). Similarly, bert-base-uncased presents a significant sensitivity to attachment type and no other effect, again in line with qualitative contrasts revealing an almost absolute favoring of LA. It is also interesting to observe that although monolingual and multilingual bert behave similarly in their striking preference for a LA interpretation across conditions, bert-base-uncased shows a slight increase of HA choices in the SC-condition, while multilingual-bert presents the same increase in the RC-only condition. Notably, neither of these patterns is human-like. In contrast, qualitative comparisons for the xlm model a tendency towards HA, stronger in the RC-only condition. For roberta, qualitative contrasts also showcase a strong, almost absolute HA preference across all conditions, with no effects of Noun-type.

C.1.4 Pairwise Comparisons by Set

Finally, even though in the main analyses of the paper we accounted for lexical effects by including random intercept for set in our statistical models, here we plot pairwise comparisons by set in order to (preliminary) evaluate model sensitivity to lexical variation. In future work, this preliminary step could be used to evaluate potential causes for the observed variation (e.g. by exploring whether a model's expectation for RC or PRs are tied to the frequency of the main clause verbs and/or the target verbs), potentially connected to corpus analyses.

C.1.5 Pairwise Comparisons: Final Notes

Across all experiments, we found some differences between statistically significant results for raw surprisal analyses, and qualitative trends in forced-choice contrasts. Davis and van Schijndel (2020) also presented both kinds of evaluations, but in their case the two methodologies overlapped with respect to the overall trend of the results. While some of the trends highlighted in the forced choice contrasts might warrant further investigation on the effect of task type, it is worth stressing that in our results these trends seem to arise from really small numerical differences in surprisal values in the item-by-item comparisons — which thus do not result in significant differences at the global level. Therefore, we must be cautious in over-interpreting these qualitative effects as overstating/emphasizing small numerical differences in values.

C.2 Prompting

While past work has leveraged prompting, in the main paper we privileged surprisal-based analyses since prompting tasks rely on specific meta-linguistic information that might confound the evaluation of model sensitivity to linguistic properties. Another issue with prompting methods lies in their generalizability across architectures — for instance, the differences in training objectives for the various LLMs tested in this paper (all of which are not considered conversational models), makes it complicated to adopt prompting as a consistent evaluation strategy. Nonetheless, we attempted to implement a prompting-based evaluation pipeline similar to Lee et al. (2025). However, while the models tested by Lee et al. (2025) were able to correctly identify which parts of a sentence corresponded to a RC and a possible DP for the RC to attach to 86 percent of the time, in our preliminary tests the three Italian models were able to identify a DP correctly only in 5.2% of the cases (on average across the three models). Since correct identification of RCs and candidate DPs is a preliminary first step in Lee et al. (2025)'s approach to investigating ambiguity resolution, failure on this constitutes an immediate roadblock in the attempt to replicate a prompt-based approach to attachment biases within the models investigates here *and* for any future attempt to conduct cross-architectural comparisons of model performance.

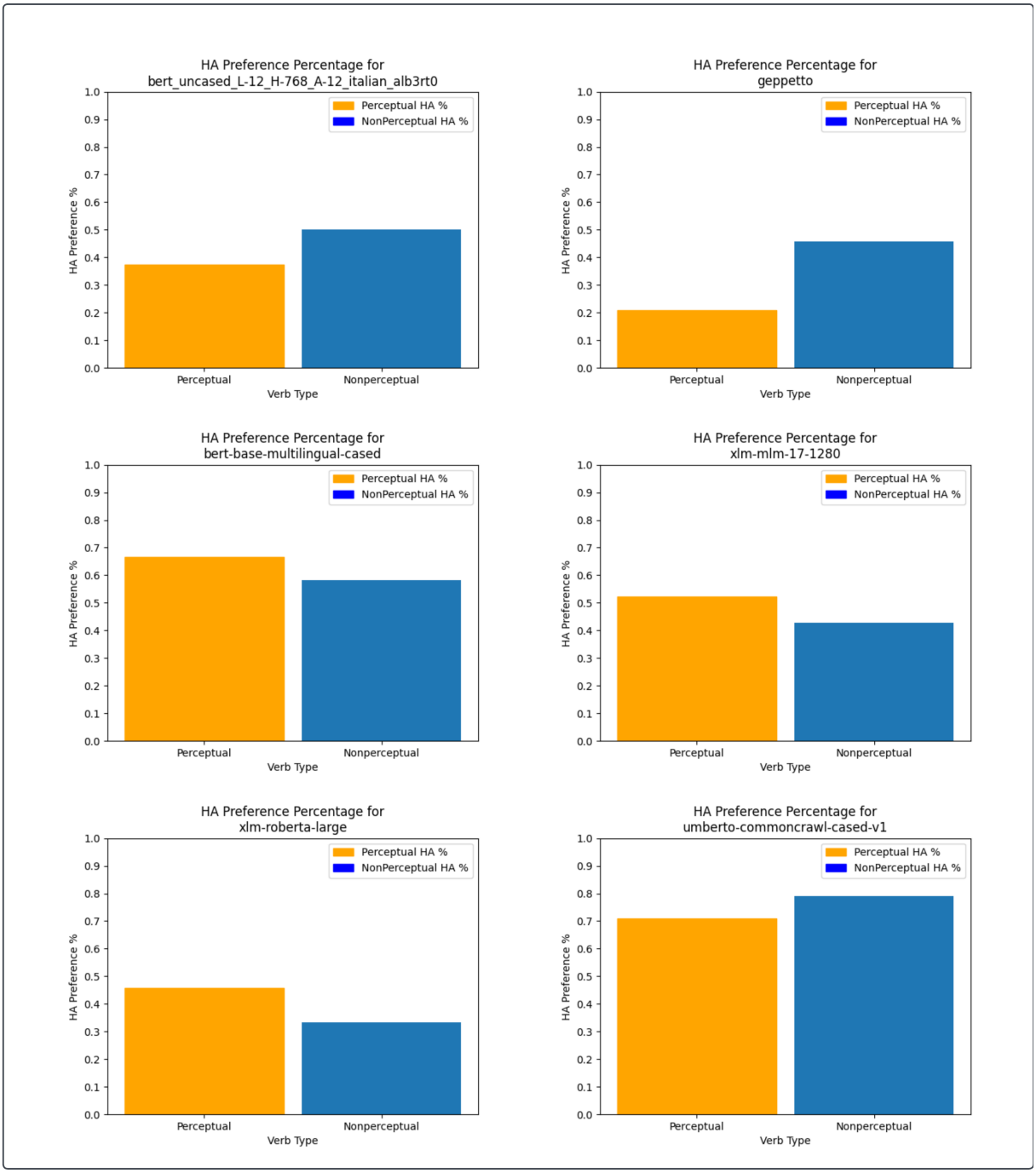


Figure 4: Proportion of HA vs. LA in the Italian Experiment, derived from categorical pairwise comparisons within sets.

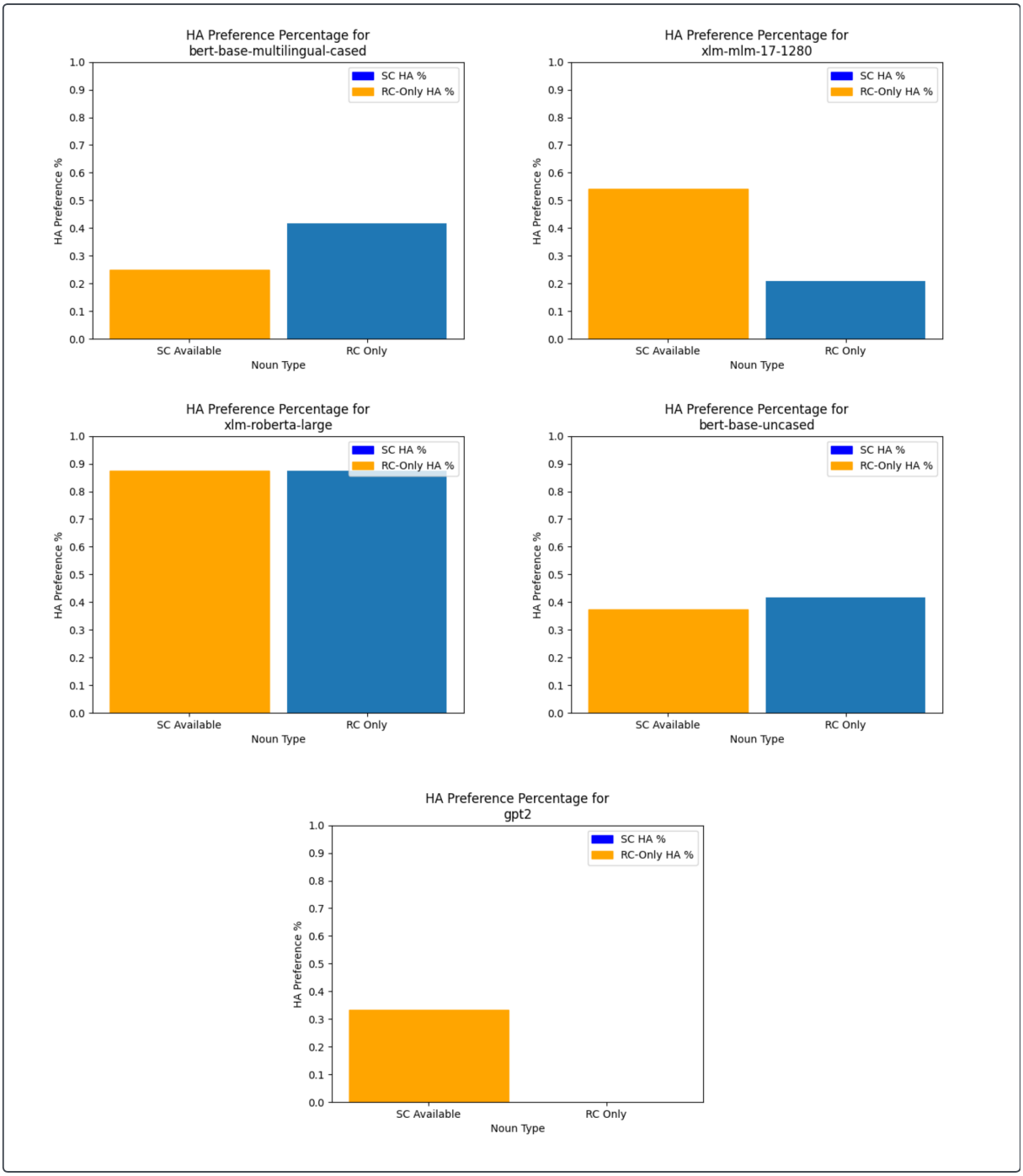


Figure 5: Proportion of HA vs. LA in the English Experiment 1, derived from categorical pairwise comparisons within sets.

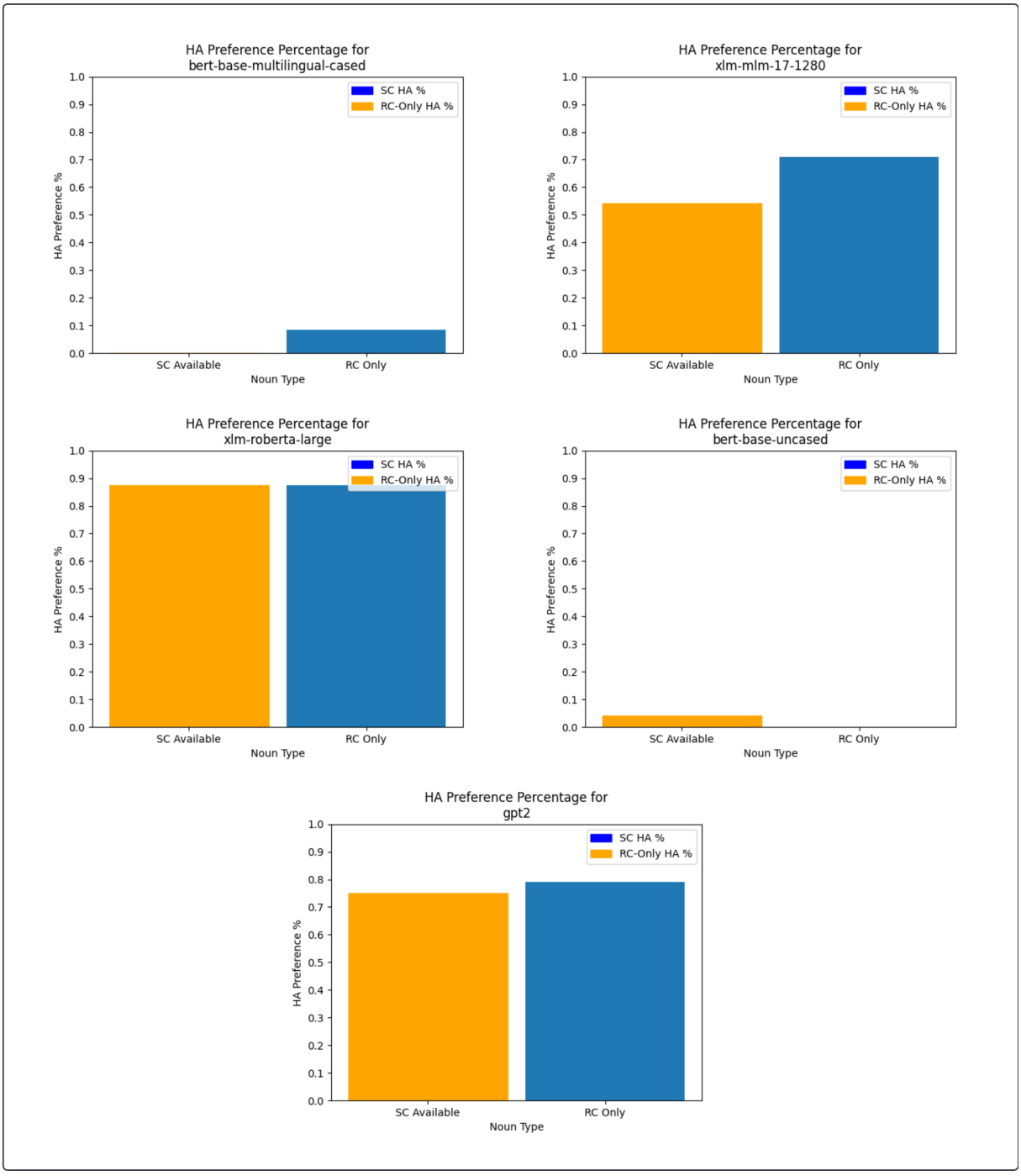


Figure 6: Proportion of HA vs. LA in the English Experiment 2, derived from categorical pairwise comparisons within sets.

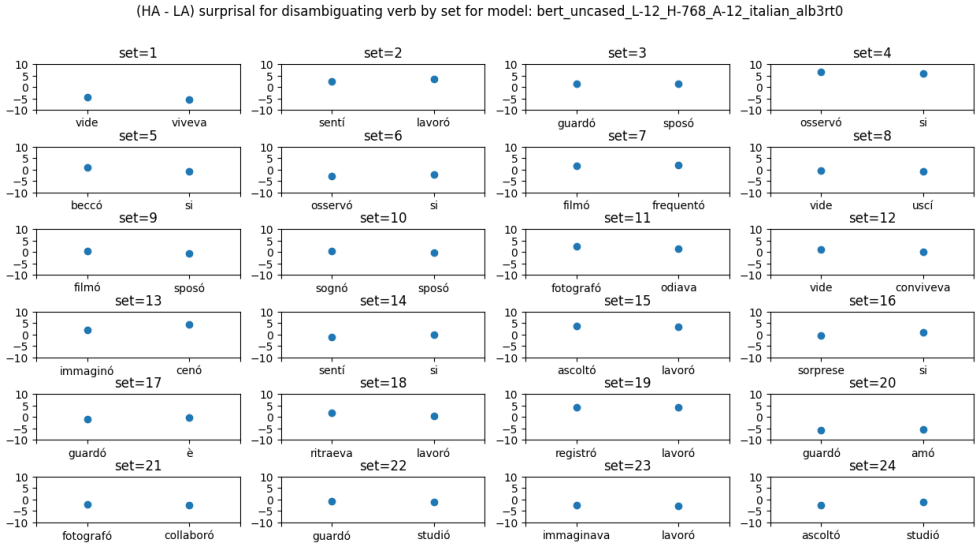


Figure 7: Surprisal Comparisons by Set and Verb Type (Perceptual vs. NonPerceptual) for alb3rt0.

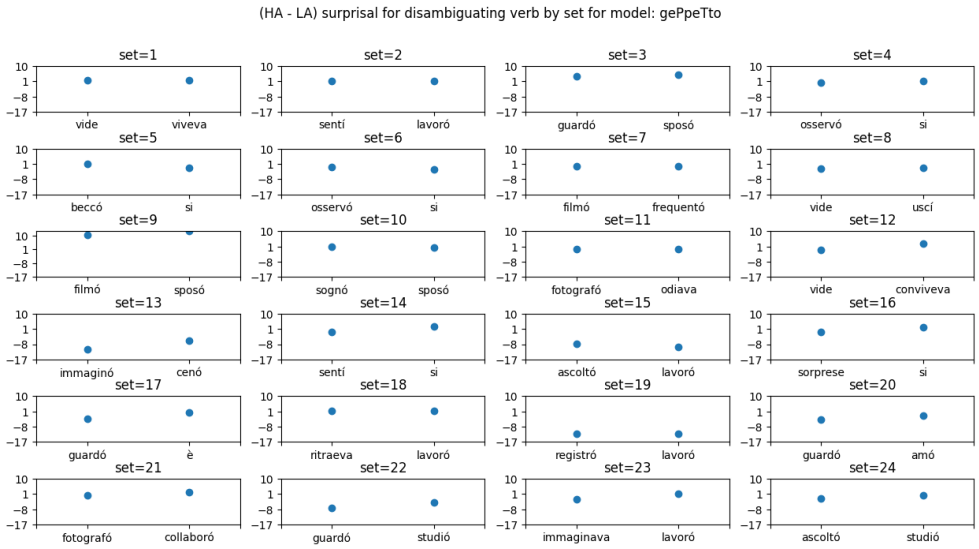


Figure 8: Surprisal Comparisons by Set and Verb Type (Perceptual vs. NonPerceptual) for gePpeTto.

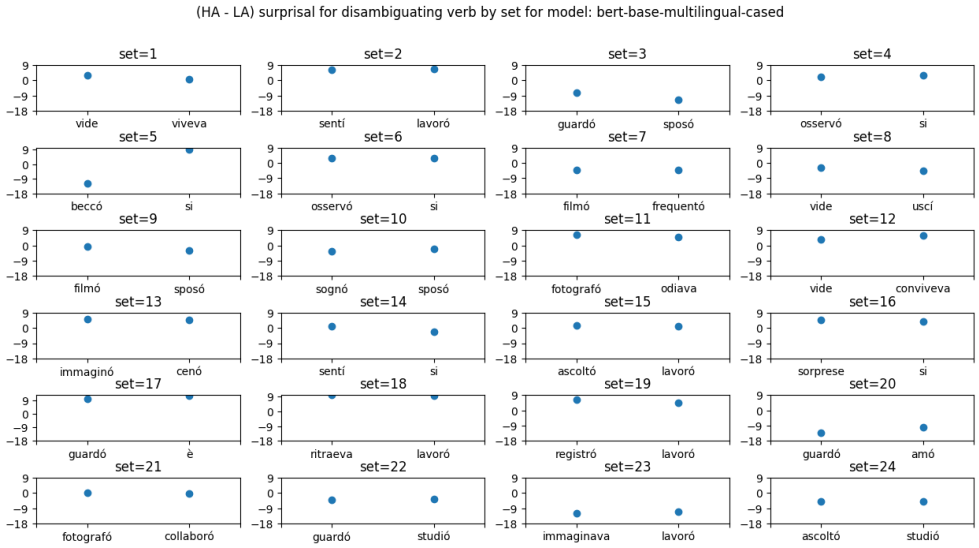


Figure 9: Surprisal Comparisons by Set and Verb Type (Perceptual vs. NonPerceptual) for Italian bert-base-multilingual-cased.

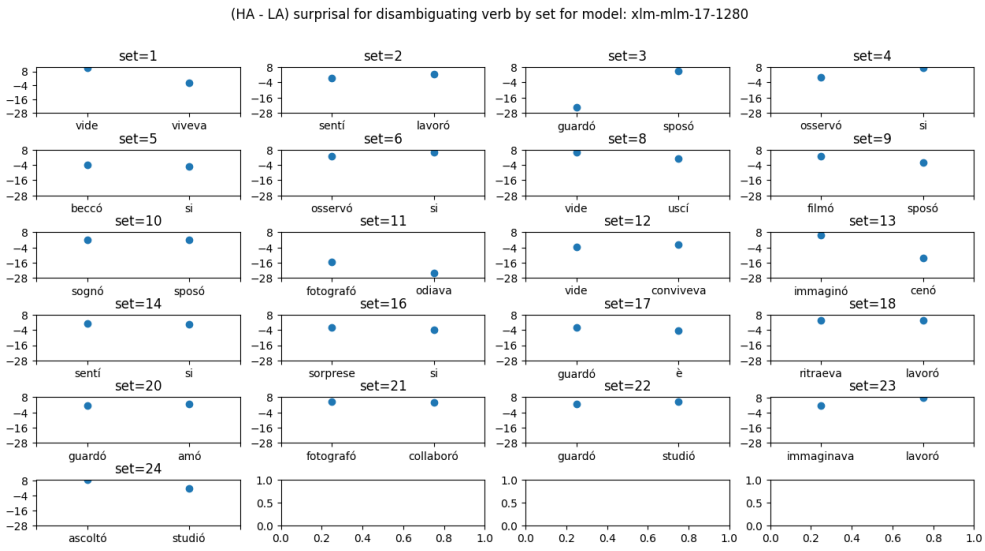


Figure 10: Surprisal Comparisons by Set and Verb Type (Perceptual vs. NonPerceptual) for xlm-mlm-17-1280

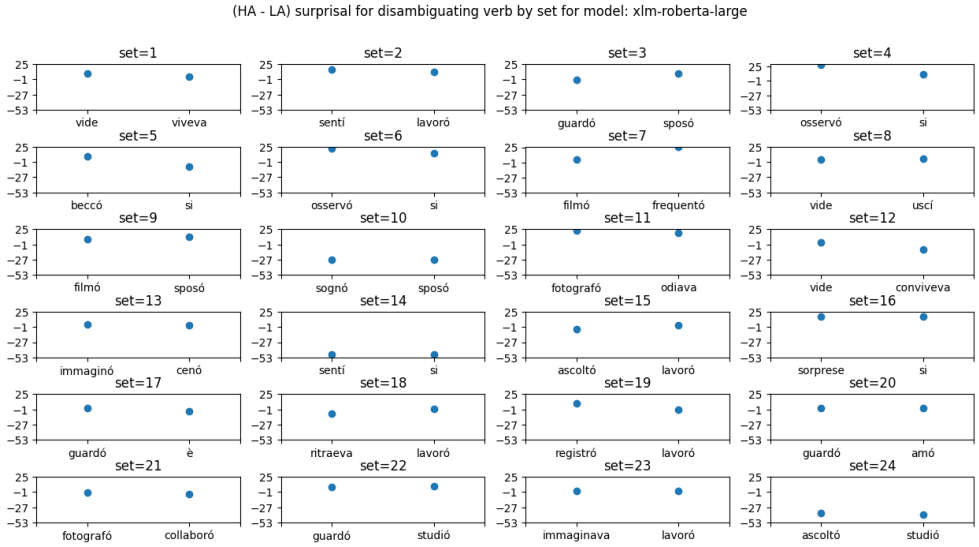


Figure 11: Surprisal Comparisons by Set and Verb Type (Perceptual vs. NonPerceptual) for Italian roberta.

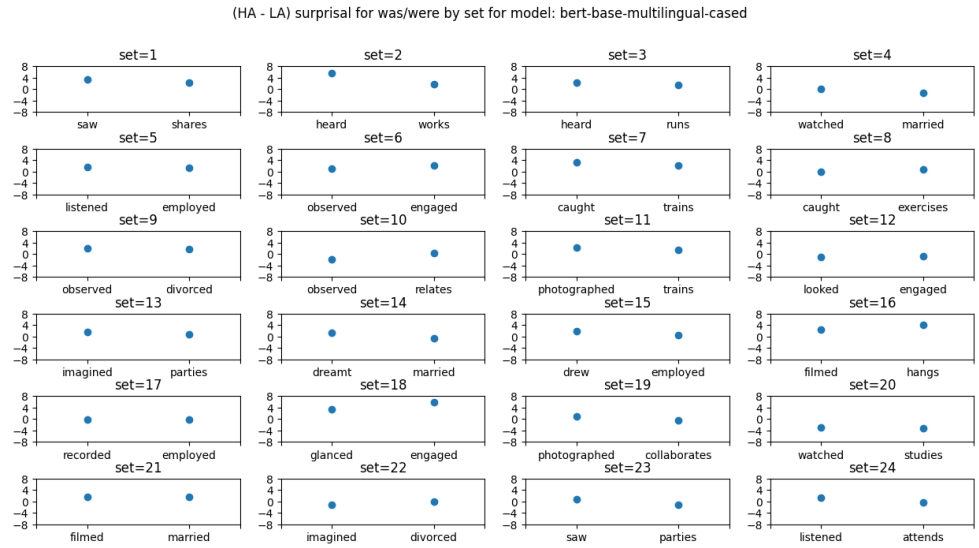


Figure 12: Surprisal Comparisons by Set and Verb Type (SC vs. RC Only) for English bert-base-multilingual-cased.

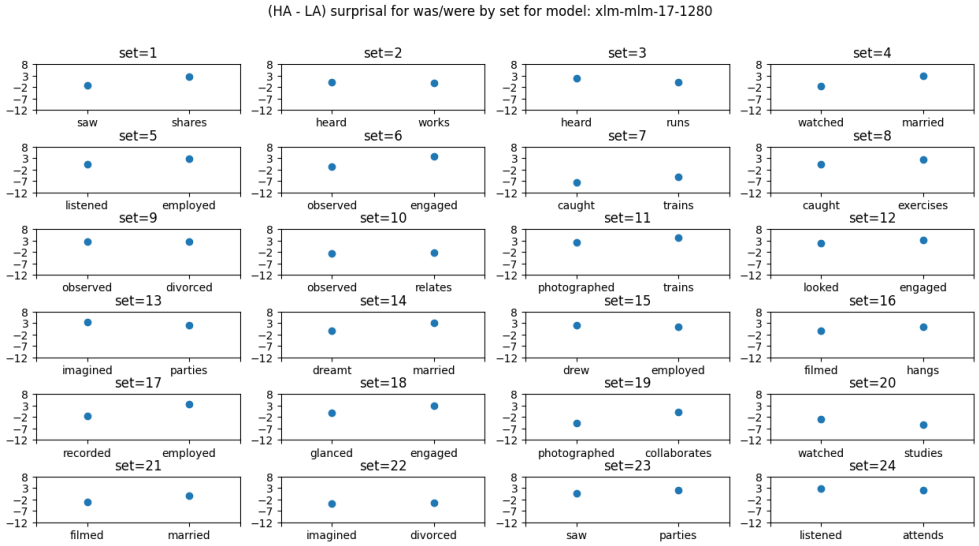


Figure 13: Surprisal Comparisons by Set and Verb Type (SC vs. RC Only) for English xlm-mlm-1280.

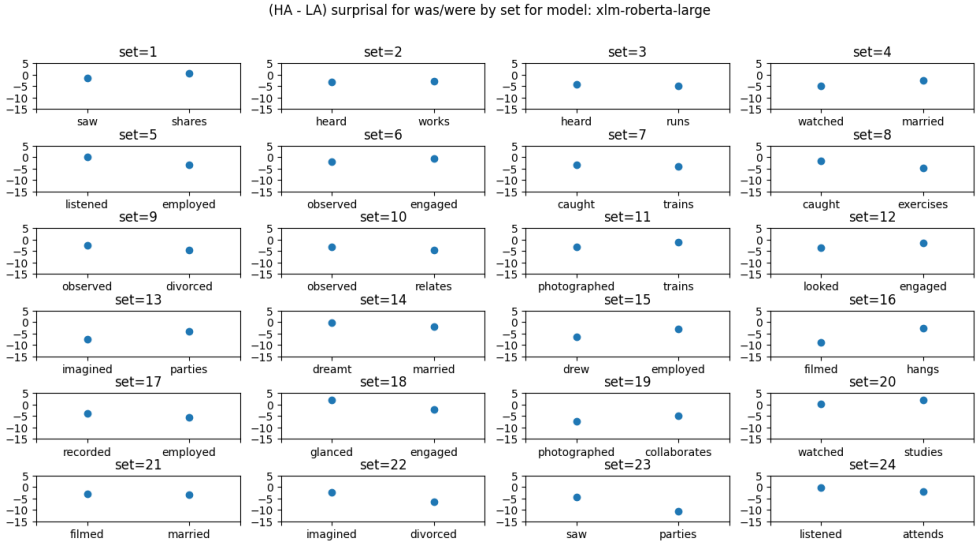


Figure 14: Surprisal Comparisons by Set and Verb Type (SC vs. RC Only) for English roberta.

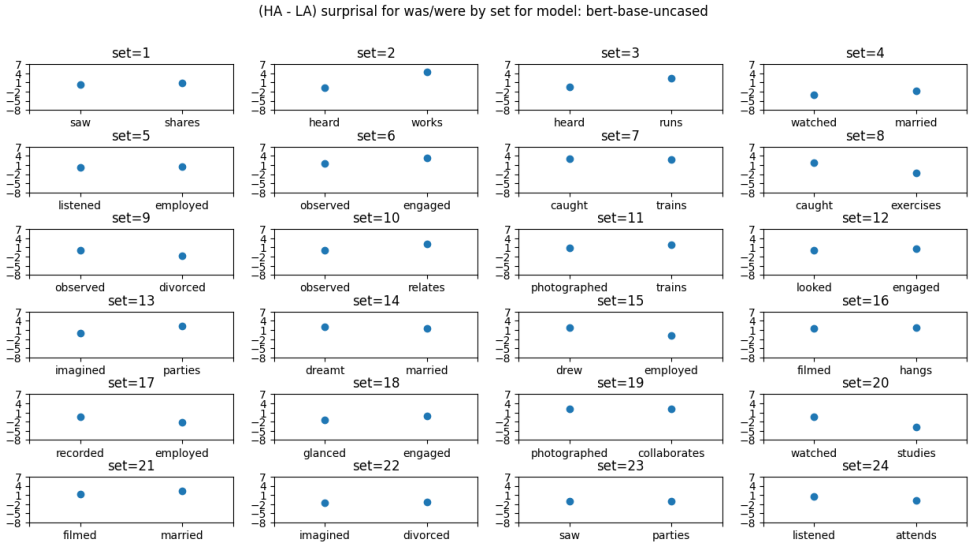


Figure 15: Surprisal Comparisons by Set and Verb Type (SC vs. RC Only) for English bert-base-uncased.

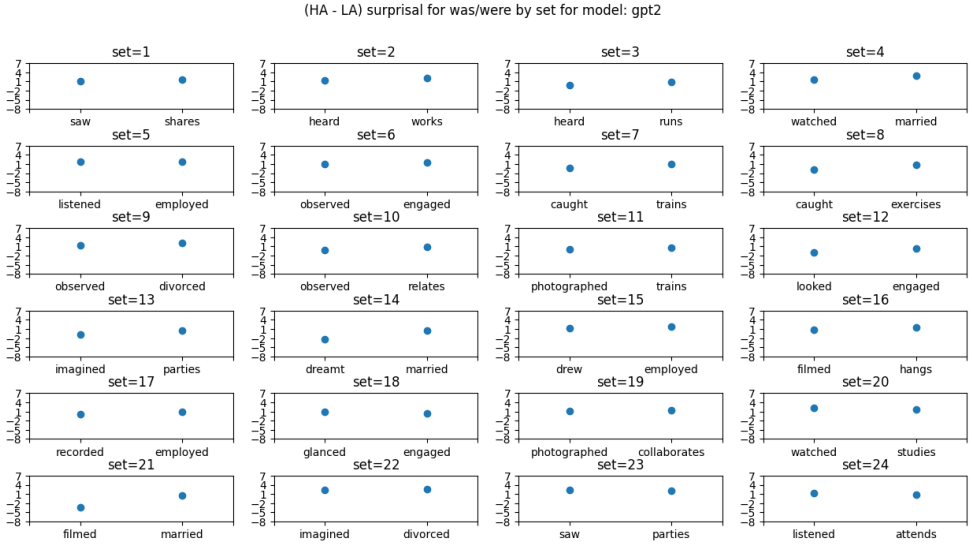


Figure 16: Surprisal Comparisons by Set and Verb Type (SC vs. RC Only) for English gpt2.

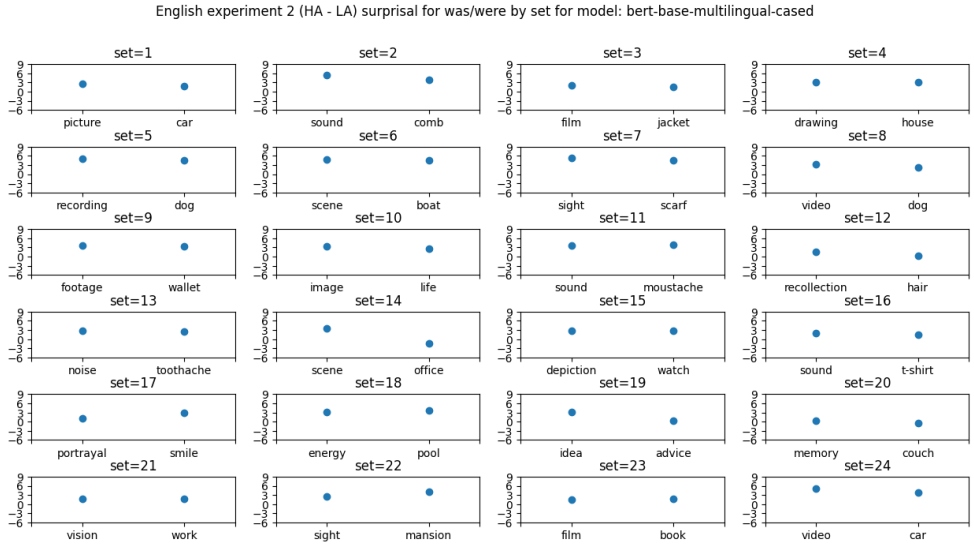


Figure 17: Surprisal Comparisons by Set and Noun Type for English bert-base-multilingual-cased.

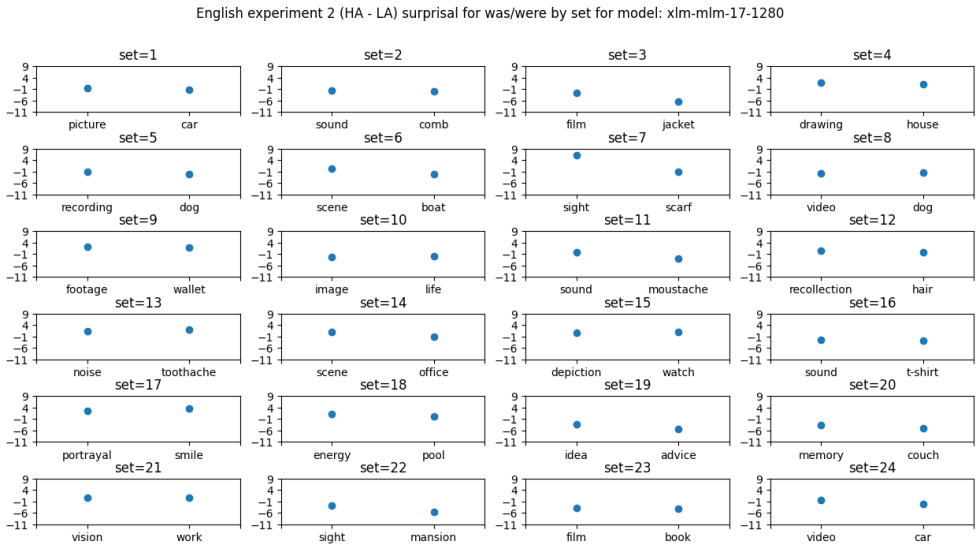


Figure 18: Surprisal Comparisons by Set and Noun Type for English xlm-mlm-17-1280.

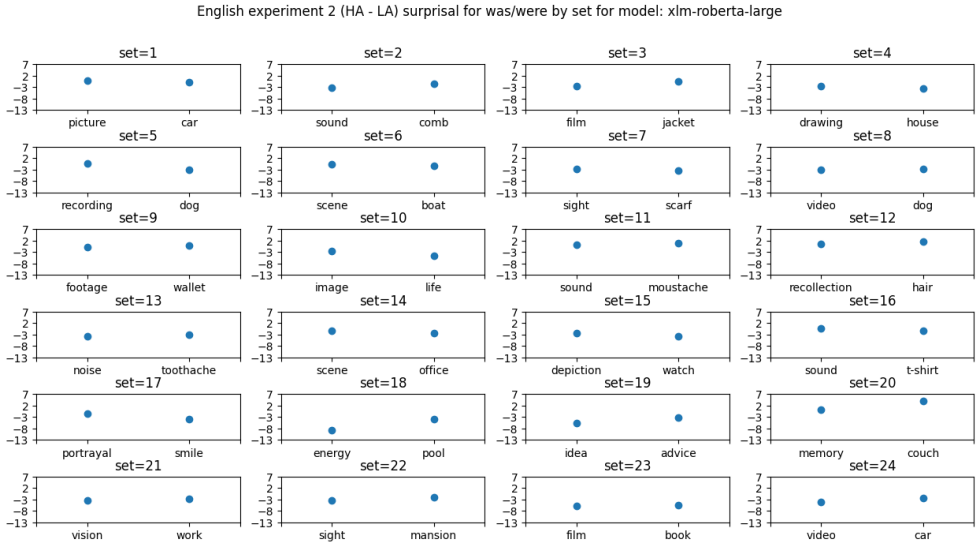


Figure 19: Surprisal Comparisons by Set and Noun for English roberta.

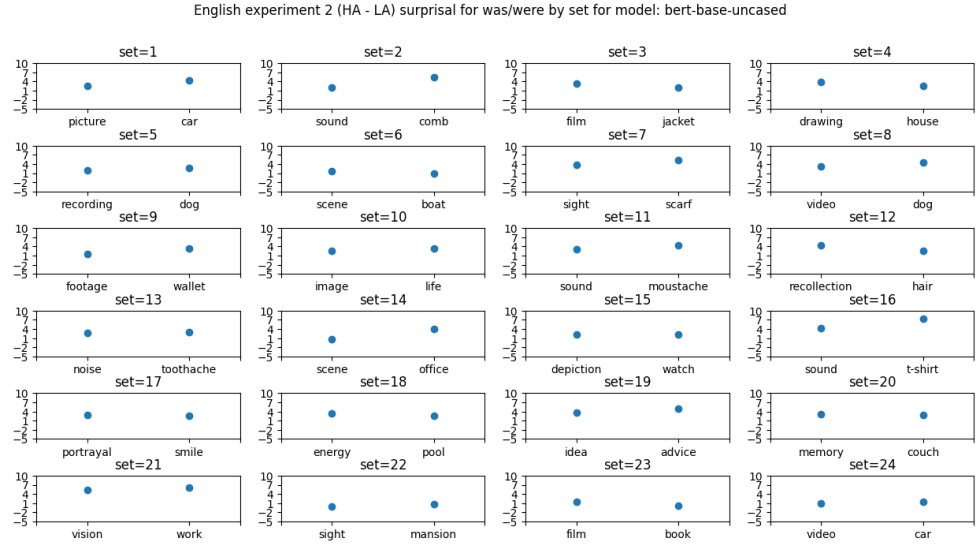


Figure 20: Surprisal Comparisons by Set and Noun Type for English bert-base-uncased.

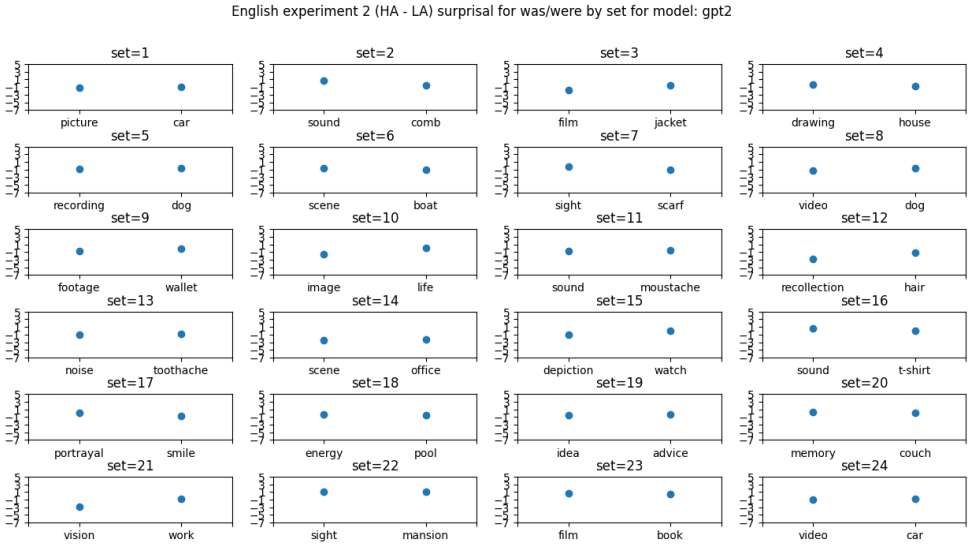


Figure 21: Surprisal Comparisons by Set and Noun Type (for English gpt2.

A Feature-Driven Tensor Semantics for Minimalist Grammars

John Paulson
Dept. of Linguistics
University of Utah
john.paulson@utah.edu

Aniello De Santo
Dept. of Linguistics
University of Utah
aniello.desanto@utah.edu

Jon Rawski
Dept. of Linguistics
& Language Development
San Jose State University
jon.rawski@sjsu.edu

Abstract

This paper shows how tensor-based distributional semantics can be incorporated into Minimalist Grammars (MGs). We embed the MG feature calculus with a tensor algebra and give a joint tensor-based representation where compositional semantics is guided by the minimalist syntax. By bridging syntactic and semantic operations in tensor spaces, we aim to contribute to the broader enterprise of neurosymbolic approaches to linguistic cognition.

1 Introduction

Many theories in formal semantics put *compositionality* at the center, in order to map complex linguistic expressions to truth values based on the semantics of their subparts (Lambek, 1958; Montague, 1970; Heim and Kratzer, 1998; Coppock and Champollion, 2025). On the other hand, implementations of the *distributional hypothesis* (Harris, 1954), i.e. approximating the meaning of expressions through distributions estimated via their co-occurrences in corpora, have been successful in computational linguistics (Erk, 2012; Clark, 2015; Lenci, 2018). By encoding distributional information in a high-dimensional vector spaces, distributional approaches to lexical semantics are easily deployable, and have become the de-facto standard as models of lexical meaning in many areas of (computational) linguistics (Mikolov et al., 2013; Boleda, 2020; Erk, 2022; Fodor and Suzuki, 2026, a.o.).

While attempts to extend distributional models to directly capture the meaning of more complex linguistic expressions exist, enhancing distributional lexical-semantics with some degree of compositionality would allow for the development of models that leverage insights from these two slightly disconnected traditions (see Baroni, 2013; Boleda, 2020, and references therein). Beyond potential benefit for Language Modeling applications, bridging the gap between propositional and distributional frameworks thus has the potential for demonstrating the compatibility of supposedly at-odds lines of inquiry into the nature of meaning (Baroni et al., 2014; Boleda and Herbelot, 2016), while contributing to the development of neuro-symbolic perspectives into human linguistic cognition more generally (Smolensky, 1990; Smolensky and Legendre, 2006; Tabor, 2009; Martin, 2020; Smolensky et al., 2022; Garcez and Lamb, 2023; Soulos, 2025; Quigley, 2025).

While a number of suggestions have been made for how to combine lexical vectors (Mitchell and Lapata, 2008; Clarke, 2009, a.o.), we are particularly interested in those proposals arguing that such combinatorial operations should be guided more or less explicitly by formal semantics and/or the steps of the syntactic derivations (Clark and Pulman, 2007; Coecke et al., 2010; Clarke et al., 2011, a.o.). For instance, Maillard et al. (2014) shows that compositional structure can be added to distributional semantics by integrating Combinatory Categorial Grammar (CCG; Steedman, 2001; Steedman and Baldridge, 2011) with distributed lexical representation encoded as tensors (see also Coecke et al., 2010; Grefenstette et al., 2011; Grefenstette, 2013b). The ability to combine vector space encoding with the semantic/syntactic derivations of CCGs is an important step towards bridging the two traditions, since CCG as a grammatical formalism is argued to be expressive enough to capture the full extent of human syntactic abilities, and has recently been proven insightful in studying the cognitive reality of linguistic structure (Stanojević et al., 2023; Isono, 2024; Ozaki et al., 2024).

However, a number of other highly expressive formalisms exist in the formal grammar literature. Among these, Stabler (1996)'s Minimalist Grammars (MGs) incorporate many concepts of modern generative linguistics, and have been gaining traction as a useful formalism for mathematically rigorous studies of the formal properties of syntactic

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 219–229
July 3-4, 2026 ©2026 Association for Computational Linguistics

patterns (Graf, 2022), as well as for the underpinning of models connecting syntactic representations to sentence processing results (Kobele et al., 2012; De Santo, 2020, 2025). An open question, then, is whether formalisms like MGs can accommodate tensor space semantics as straightforwardly as CCG. Crucially, some work already exists proving that the syntactic feature calculus of MGs can be reinterpreted as operations in vector spaces (beim Graben and Gerth, 2012). In this paper, we answer this question by directly leveraging beim Graben and Gerth (2012) to extend Maillard et al. (2014)'s results to MGs. Specifically, we show how to integrate tensor-based semantics into a basic version of MGs, by associating the word meanings to tensors whose semantic features match the syntactic features of the corresponding MG lexical items. Like the CCG case, the core intuition is that since tensors are multilinear maps, they can be directly manipulated using the feature-driven combinatorial calculus of MGs.

2 Mathematical Preliminaries

Here we overview some basic notions of tensors and multilinear algebra (see Kolda and Bader, 2009; Lang, 2002, for more details). We assume familiarity with vector spaces and linear algebra.

For finite-dimensional real vector spaces V, W, the *tensor product* $V \otimes W$ is the vector space spanned by $v \otimes w = vw^T$ for all $v \in V$, $w \in W$, i.e. it is a generalization of the outer product. Iterating this product yields a vector space $V_1 \otimes \cdots \otimes V_k$, whose elements are *order-k tensors*. Order-k tensors can also be considered as a k-dimensional array $\mathcal{T} \in \mathbb{R}^{d_1 \times \cdots \times d_k}$, as they can be fully described using k indices (i.e. rows, columns, etc). Order-0, order-1, and order-2 tensors correspond to scalars, vectors, and matrices, respectively.

Much like the outer product creates tensors of a higher order, the inner product $v \cdot w = v^T w$ decreases, or *contracts* tensor order. *Tensor contraction* is a generalization of the inner product to tensors of any order. Contracting order-2 and order-1 tensors is matrix-vector multiplication and yields a vector, contracting two order-2 tensors is matrix multiplication, yielding a matrix, and so on.

For a matrix $M \in W \otimes V$ and vector $v \in V$, we say the (right) contraction $M \cdot v \in W$ is $M_{ij} v_j$. In general, we say for a tensor $\mathcal{T} \in V_k \otimes \cdots \otimes V_1$ and a vector $v \in V_1$, the contraction $\mathcal{T} \cdot v \in V_k \otimes \cdots \otimes V_2$. Iterating such contractions will always mean right iteration, with the usual convention that repeated indices are implicitly summed. As a notational convenience when contracting, for instance, tensors A_{ijk} and B_k we sometimes write $A_{ijk} \cdot B_k = A_{ij}$ (i.e., $A_{ij} = \sum_k A_{ijk} B_k$).

There is an isomorphism between tensors and multilinear maps: a map $f : V_1 \to \ldots \to V_j \to V_k$ can be represented as a tensor $\mathcal{T}^f \in V_k \otimes V_j \otimes \ldots \otimes V_1$. This means that tensor contraction acts as function application, i.e. for any $v_1 \in V_1, \ldots, v_j \in V_j$, $f v_1 \ldots v_j = v_k = \mathcal{T}^f \cdot v_1 \cdot \ldots \cdot v_j$.

3 CCG and Tensor Semantics

Our approach to incorporating tensor-based semantics with MGs draws heavily from previous results on CCG. In this section, we give an intuitive sketch of CCG's type-driven calculus, and we walk the reader through an example of how it can be tied to tensor-based semantics. The reader is referred to Steedman and Baldridge (2011) and Maillard et al. (2014) for more detailed introductions to CCGs and to vector space operations over CCG derivations, respectively.

CCG (Steedman, 2001) is a radically lexicalized, mildly-context sensitive grammar formalism in which syntactic types (categories) identify syntactic objects as *primitive categories* or *functions* (Steedman and Baldridge, 2011, p. 184). Primitive categories are those such as noun (N), noun phrase (NP), sentence (S), etc. Functions take arguments of a particular category (either a primitive or function) and outputs something of another category. Thus, function types identify the category of their output, the category of their argument(s), as well as the order in which the arguments must combine. For instance, A/B and $A\backslash B$ are functions that take an element of category B (to the right and to the left, respectively) and return an element of category A. As a more concrete example, transitive verbs can be defined by a syntactic category $(S\backslash NP)/NP$.

Primitive categories and functions can be combined through a set of syntactic rules, which include *functional application* of functions to arguments, as well as combinatory rules like *type-raising* and *functional composition*. Crucially, the application of syntatic rules is fully dependent on the categories of the lexical items involved in the derivation.

For the sake of illustration, let us briefly exemplify functional application. Consider a lexicon with three lexical items: *Alice* and *Bob* both with category NP, and *sees*, with category $(S\backslash NP)/NP$. We can then derive *Alice sees Bob*

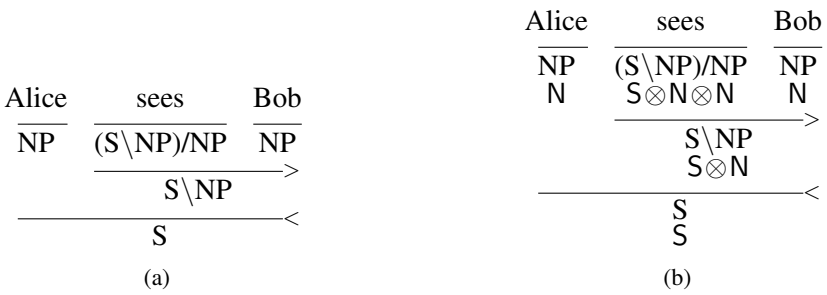


Figure 1: A CCG derivation for the sentence "Alice sees Bob", and the same derivation, with the tensor-semantics shown for each substructure.

by first combining *sees* with *Bob* to produce the expression *sees Bob* (a function of type $S\backslash NP$). Then, this function can combine with *Alice* to yield *Alice sees Bob* with type S (see Figure 1a).

Importantly, one of the core features of CCG is that it commits to a a transparent interface between syntax and semantics, in that the semantic type of a linguistic expression is entirely determined by the syntactic category. Therefore, the combinatorial rules illustrated above perform syntactic and semantic composition in parallel. In turn, this tight link between syntactic and semantic types seemingly makes CCG uniquely suited to the integration of combinatorial operations with vector-based semantics. In this sense, Maillard et al. (2014) exemplifies how to perform tensor-based operations in parallel with the core combinatorial rules of CCG (Coecke et al., 2010; Grefenstette, 2013a). Essentially, each primitive category is associated to a corresponding (distinct) vector space, e.g. S to S and NP to N. Suppose *Alice* and *Bob* have category NP, as before. The corresponding vectors in N would be A_j and B_k.

The vector spaces associated to functions' more complex types are determined through the tensor product of the spaces of the primitive categories each type is composed of. That is, each primitive type is associated with a corresponding vector space, and the slash operators are replaced with the tensor product operator. For instance, the transitive verb category $(S\backslash NP)/NP$ is associated to the vector space $\mathsf{S}\otimes\mathsf{N}\otimes\mathsf{N}$. Thus, the transitive verb *sees* has tensor $E_{ijk}\in\mathsf{S}\otimes\mathsf{N}\otimes\mathsf{N}$, with a distinct index for each component's vector space.

When function application happens in CCG, the corresponding tensors are contracted. In the first step of the derivation for *Alice sees Bob*, we get the tensor $E_{ijk}\cdot B_k = E_{ij}\in\mathsf{S}\otimes\mathsf{N}$, and then $E_{ij}\cdot A_j = E_i\in\mathsf{S}$ (see Figure 1b). Maillard et al. (2014) show how the other combinatory rules map to vector space operations in similarly straighforward ways.

4 Minimalist Grammars and Tensor Representations

The goal of this paper is to incorporate tensor-based semantic representations with Minimalist Grammars. In the previous section, we outlined the first piece of the puzzle by showing how tensor-semantics can be tied to CCG-like categories and type-driven combinatorics. We now turn to our grammar formalism of interest, sketching the intuition behind its structure-building operations and how the syntactic calculus can be mapped to vector space operations.

MGs are a derivational, lexicalized, and feature-driven mildly-context sensitive formalism able to express the complex structural analyses of generative syntax (Stabler, 1996, 2013). Intuitively, an MG grammar consists of a set of lexical items associated with a non-empty, finite string of syntactic features. Lexical items and more complex syntactic objects can be combined via two feature-driven operations — merge and move.[1] CCG-style complex categories for syntactic objects are represented in MG simply by an item's feature string. In this sense, MGs are type-driven: all of the information that is input to the combinatorial rules is contained within the feature strings themselves.

For convenience, the features relevant for the Merge and Move operations are standardly taken to be different. With respect to Merge, each lexical item will contain one simple categorical feature acting as the selectee feature (denoted f), and 0 or more selector features (denoted $=\mathrm{f}$). With respect to Move, a lexical item might contain 0 or more licensee features (denoted f^-) and licensor features

[1] We set aside extensions to MG that accommodate the Adjoin operation (Frey and Gärtner; Fowlie, 2013), which is also not included in beim Graben and Gerth (2012)'s geometric MG formalization.

(denoted f^{+}). The operation Merge then takes two structures, one whose feature string begins with a basic category (f) feature, and one whose string begins with a selector (=f) feature, combines them, and deletes both features. Move operates similarly with licensor (f^{+}) and licensee (f^{-}) features, except that the structure with the targeted licensee feature is already a substructure of the one with the licensor. Thus, Merge is a binary operation encoding subcategorization, while Move is a unary operation allowing for the re-ordering of already incorporated items.

In this sense, MG derivations can be interpreted as acting on term algebras of trees and feature arrays, using two partial operations:

$$\begin{aligned} \text{merge} &: T_A \times T_A \rightharpoonup T_A, \\ \text{move} &: T_A \rightharpoonup T_A, \end{aligned}$$

where T_A is the set of well-formed trees over a fixed minimalist grammar G, and whose domains are $\mathrm{dom}(\text{merge})$ and $\mathrm{dom}(\text{move})$, respectively (beim Graben and Gerth, 2012).

Importantly, for each lexical item, only one feature is *accessible* to merge or move at each step: the currently accessible feature is the feature at the beginning (left-most) position of the feature string. This implies that some features will only available for checking after others have been checked/deleted, based on their order in their respective feature strings. As an illustrative example, Figure 2 shows an MG derivation for the sentence "*What did Alice eat*", under standard Minimalist assumptions for the structure of the derivation (Koopman and Sportiche, 1991; Adger, 2003). Note how MGs allow for phonologically empty (Hornstein et al., 2005) but syntactically relevant lexical items, as common in many versions of minimalist theories.

Crucially, beim Graben and Gerth (2012) show that the operation merge and move over feature-strings can be mapped to vector space operations acting over tensor-product representations tied to structural positions over MG derivation trees. Specifically, it is possible to define a vector space $\mathscr{F}$, an embedding $\boldsymbol{\Psi} : T_A \to \mathscr{F}$, and partial operations

$$\begin{aligned} \textbf{merge} &: \mathscr{F} \times \mathscr{F} \rightharpoonup \mathscr{F} \\ \textbf{move} &: \mathscr{F} \rightharpoonup \mathscr{F} \end{aligned}$$

whose domains are $\mathrm{dom}(\textbf{merge})$ and $\mathrm{dom}(\textbf{move})$, respectively. Notationally, merge and move are the two structure-building operations on the tree term algebra and **merge** and **move** their corresponding

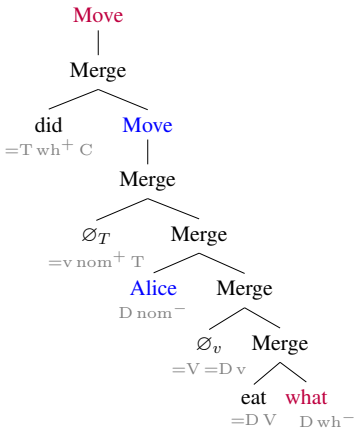


Figure 2: An MG derivation of the sentence "*What did Alice eat*". On each leaf, we show in lighter gray the feature string associated to a lexical item below its corresponding phonological content.

operations on the tensor algebra. beim Graben and Gerth (2012) prove that this mapping forms a homomorphism between the two term algebras:

Theorem 4.1 (beim Graben and Gerth (2012), Thm. 1)**.** $\boldsymbol{\Psi}$ is a homomorphism from $(T_A, \text{merge}, \text{move})$ to $(\mathscr{F}, \textbf{merge}, \textbf{move})$, where for $(t_1, t_2) \in \mathrm{dom}(\text{merge})$ and $(t) \in \mathrm{dom}(\text{move})$, $\boldsymbol{\Psi}(\text{merge}(t_1, t_2)) = \textbf{merge}(\boldsymbol{\Psi}(t_1), \boldsymbol{\Psi}(t_2))$ and $\boldsymbol{\Psi}(\text{move}(t)) = \textbf{move}(\boldsymbol{\Psi}(t))$.

We directly leverage this result in our construction of an MG tensor-based semantics.

5 A Tensor Semantics for MGs

With this background in place, we get to the core of this paper's contribution. We now show how to enhance distributional semantics with compositional operations guided by the structure-building steps of MGs. Our construction proceeds in analogy to Maillard et al. (2014), where syntactic features determine the semantic tensor types (Kobele, 2006) and semantic composition is transparently reflected in the syntactic composition (Hunter, 2007; Kobele, 2012; Tomita, 2016) via tensor contraction (beim Graben and Gerth, 2012; Maillard et al., 2014).

We consider an MG with a finite set $\mathcal{S}$ of selectee features and a corresponding lexicon Lex. Following Maillard et al. (2014), who assigned a vector space for basic CCG categories, we assign a vector space f for each selectee feature $\mathrm{f} \in \mathcal{S}$, and, using the direct sum, let $\mathsf{S} := \bigoplus_{\mathrm{f} \in \mathcal{S}} \mathsf{f}$. We then embed every lexical item in $\mathsf{L} := \bigoplus_{k \geq 1} \mathsf{S}^{\otimes k}$, where $\mathsf{S}^{\otimes k}$ is the tensor product of S with itself k times. Since selector features are checked left-to-right, their

corresponding space is in reverse order in the tensor product, so the currently active selector always corresponds to the rightmost tensor contraction.

Definition 5.1. A *lexical embedding* is a map $\varphi : Lex \to \mathsf{L}$ such that for each lexical item $l \in Lex$, if l's feature string contains selectee feature $\mathrm{f} \in \mathcal{S}$ and selector features $=\mathrm{f}_1,\ldots,=\mathrm{f}_k$, then $\varphi(l) \in \mathsf{f} \otimes \mathsf{f}_k \otimes \cdots \otimes \mathsf{f}_1 \subseteq \mathsf{L}$.

Note that φ maps selector features (e.g. $=\mathrm{N}$) to the same space of the "corresponding" selectee feature (e.g. N). We can extend this to a general compositional semantics over derivation trees.

Definition 5.2. Given a lexical embedding φ, we define $\boldsymbol{\Phi} : T_A \to \mathsf{L}$ as:

$$\begin{aligned} \boldsymbol{\Phi}(l) &= \varphi(l) \\ \boldsymbol{\Phi}(\mathrm{move}(t)) &= \boldsymbol{\Phi}(t) \\ \boldsymbol{\Phi}(\mathrm{merge}(t_1,t_2)) &= \boldsymbol{\Phi}(t_1) \cdot \boldsymbol{\Phi}(t_2) \end{aligned}$$

where $l \in Lex$, $t \in \mathrm{dom}(\mathrm{move})$, $(t_1, t_2) \in \mathrm{dom}(\mathrm{merge})$.

We note that this function $\boldsymbol{\Phi}$ is unique. We also note that (right) contraction $(\cdot)$ works on the current selector/selectee pair, so is well-typed for $\boldsymbol{\Phi}(t_1), \boldsymbol{\Phi}(t_2)$ when $(t_1, t_2) \in \mathrm{dom}(\mathrm{merge})$.

We now combine the syntactic embedding $\boldsymbol{\Psi}$ with the semantic embedding $\boldsymbol{\Phi}$ into a paired representation.

$$\begin{aligned} \boldsymbol{\Omega} : T_A &\to (\mathscr{F} \times \mathsf{L}) \\ t &\mapsto (\boldsymbol{\Psi}(t), \boldsymbol{\Phi}(t)) \end{aligned}$$

Furthermore, we define the partial functions $\mathcal{M}erge$ and $\mathcal{M}ove$:

$$\begin{aligned} \mathcal{M}erge &: (\mathscr{F} \times \mathsf{L}) \times (\mathscr{F} \times \mathsf{L}) \rightharpoonup \mathscr{F} \times \mathsf{L} \\ &\quad ((a,A),(b,B)) \mapsto (\mathbf{merge}(a,b), A \cdot B) \\ \mathcal{M}ove &: \mathscr{F} \times \mathsf{L} \rightharpoonup \mathscr{F} \times \mathsf{L} \\ &\quad (a,A) \mapsto (\mathbf{move}(a), A) \end{aligned}$$

Following beim Graben and Gerth (2012), we show this "joint" embedding is a homomorphism.

Proposition 5.3. $\boldsymbol{\Omega}$ is a homomorphism from $(T_A, \mathrm{merge}, \mathrm{move})$ to $((\mathscr{F} \times \mathsf{L}), \mathcal{M}erge, \mathcal{M}ove)$, where for $(t_1, t_2) \in \mathrm{dom}(\mathrm{merge})$ and $t \in \mathrm{dom}(\mathrm{move})$, $\mathcal{M}erge(\boldsymbol{\Omega}(t_1), \boldsymbol{\Omega}(t_2)) = \boldsymbol{\Omega}(\mathrm{merge}(t_1,t_2))$ and $\mathcal{M}ove(\boldsymbol{\Omega}(t)) = \boldsymbol{\Omega}(\mathrm{move}(t))$.

Proof. Assume $(t_1, t_2) \in \mathrm{dom}(\mathrm{merge})$. By Thm 4.1 and Def 5.2, $\mathbf{merge}(\boldsymbol{\Psi}(t_1), \boldsymbol{\Psi}(t_2)) = \boldsymbol{\Psi}(\mathrm{merge}(t_1, t_2))$ and $\boldsymbol{\Phi}(\mathrm{merge}(t_1, t_2)) = \boldsymbol{\Phi}(t_1) \cdot \boldsymbol{\Phi}(t_2)$. Therefore $\mathcal{M}erge(\boldsymbol{\Omega}(t_1), \boldsymbol{\Omega}(t_2)) = \big(\boldsymbol{\Psi}(\mathrm{merge}(t_1, t_2)), \boldsymbol{\Phi}(\mathrm{merge}(t_1, t_2))\big) = \boldsymbol{\Omega}(\mathrm{merge}(t_1,t_2))$. The move case is analogous. □

Our MG tensor semantic formulation has several nice properties. First, every merge step applies a tensor meaning to a vector argument by contraction, and so we only need tensor-vector maps. Moreover, this construction highlights the fact that MG's merge is essentially functional application (Kobele, 2006), and as such merge steps are the only ones that "add" lexical semantic content to a derivation. The relationship between move and propositional semantics is less straightforward, as move is also argued to determine scope-taking. Here, we focused on the case in which the lexical semantic content of a mover is fully integrated in the derivation at the time the mover is first merged. We return to these more complicated effects of move on the semantics of a derivation in Section 6.

5.1 A Merge-Only Example

To better illustrate the above results, we first start by exemplifying the tensor semantics calculus over a minimal MG derivations, simplifying most syntactic assumptions proper of minimalist syntax in order to only focus on our treatment of merge.

Suppose we have three lexical items (with their corresponding MG feature strings): *Alice* :: [D], *Bob* :: [D], and *sees* :: [=D, =D, C], along with their syntactic and semantic embeddings:

$$\begin{aligned} &\mathit{Alice} :: [\mathrm{D}], \\ &\boldsymbol{\Psi}(\mathit{Alice}) = \mathbf{Alice}, \\ &\boldsymbol{\Phi}(\mathit{Alice}) = A_j \in \mathsf{D} \end{aligned}$$

$$\begin{aligned} &\mathit{Bob} :: [\mathrm{D}], \\ &\boldsymbol{\Psi}(\mathit{Bob}) = \mathbf{Bob}, \\ &\boldsymbol{\Phi}(\mathit{Bob}) = B_k \in \mathsf{D} \end{aligned}$$

$$\begin{aligned} &\mathit{sees} :: [=\mathrm{D}, =\mathrm{D}, \mathrm{C}], \\ &\boldsymbol{\Psi}(\mathit{sees}) = \mathbf{sees}, \\ &\boldsymbol{\Phi}(\mathit{sees}) = G_{ijk} \in \mathsf{C} \otimes \mathsf{D} \otimes \mathsf{D} \end{aligned}$$

Thus we have $\boldsymbol{\Omega}(\mathit{Alice}) = (\mathbf{Alice}, A_j)$, $\boldsymbol{\Omega}(\mathit{Bob}) = (\mathbf{Bob}, B_k)$, $\boldsymbol{\Omega}(\mathit{sees}) = (\mathbf{sees}, G_{ijk})$. We can now derive the expression *Alice sees Bob* (see Figure 3).

Step 1. First, we merge *sees* and *Bob*, checking (deleting) the $= \mathrm{D}$ feature from *sees* and the D feature from *Bob*. This gives us a subtree for the expression *sees Bob* with MG feature string $[=\mathrm{D}, \mathrm{C}]$.

To illustrate how the syntactic and semantic derivation proceed in parallel to this in the corresponding vector spaces, recall that the vectors $\mathbf{\Omega}(\mathit{Bob})$, $\mathbf{\Omega}(\mathit{sees})$ are in the space $\mathscr{F} \times \mathsf{L}$, where $\mathscr{F}$ represents the *syntactic* component and L represents the (lexical) *semantic* component. Given our construction, when two MG structures are combined with merge, this is reflected by using the **merge** operation in $\mathscr{F}$, and by contracting the tensors in L. To get the result in $\mathscr{F} \times \mathsf{L}$, we use the $\mathcal{M}erge$ operation, which simply applies both operations to the relevant components. Therefore, we can find $\mathbf{\Omega}(\mathit{sees\ Bob})$ as follows:

$$\begin{aligned}&\mathbf{\Omega}(\mathit{sees\ Bob})\\ =&\mathcal{M}erge((\mathbf{sees}, G_{ijk}), (\mathbf{Bob}, B_k))\\ =&(\mathbf{merge}(\mathbf{sees}, \mathbf{Bob}), G_{ij})\end{aligned}$$

As a reminder, G_{ij} is defined to be the contraction $G_{ijk} \cdot B_k$. As expected since *sees Bob* has feature string $[=\mathrm{D},\ \mathrm{C}]$, $\mathbf{\Phi}(\mathit{sees\ Bob}) = G_{ij}$ is an element of $\mathsf{C} \otimes \mathsf{D}$.

Step 2. The next merge checks the remaining $=\mathrm{D}$ feature on *sees Bob* with the D feature from *Alice*, resulting in the expression *Alice sees Bob* On the tensor side, applying again the $\mathcal{M}erge$ operation, we have:

$$\begin{aligned}&\mathbf{\Omega}(\mathit{Alice\ sees\ Bob})\\ =&\mathcal{M}erge((\mathbf{merge}(\mathbf{sees}, \mathbf{Bob}), G_{ij}), (\mathbf{Alice}, A_j))\\ =&(\mathbf{merge}(\mathbf{merge}(\mathbf{sees}, \mathbf{Bob}), \mathbf{Alice}), G_i)\end{aligned}$$

Correctly, G_i, our "final product" of the semantic component, is in C — matching the remaining feature C in the MG feature string of *Alice sees Bob*. By comparing Figure 3 to Figure 1b, it should be evident how our tensor-based semantic calculus mirrors Maillard et al. (2014)'s almost exactly when we limit ourselves to MG's merge operation.

5.2 An Example with Merge and Move

We will now exemplify the tensor calculus over the complete MG derivation introduced earlier (Fig. 2). Suppose we have the following LIs from an MG, along with their images under $\mathbf{\Psi}$ and $\mathbf{\Phi}$:[2]

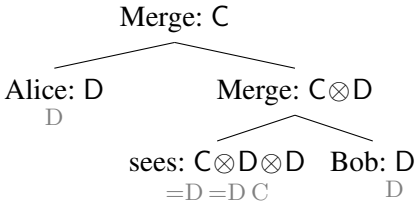


Figure 3: MG derivation for a simplified analysis of "*Alice sees Bob*", showing the semantic vector spaces corresponding to each substructure. Given our focus on the tensor semantics, we leave out the parallel tensor-syntax for ease of visualization.

$$\begin{aligned}\mathit{what} &:: [\mathrm{D}, \mathrm{wh}^-],\\ \mathbf{\Psi}(\mathit{what}) &= \mathbf{what},\\ \mathbf{\Phi}(\mathit{what}) &= W_k \in \mathsf{D}\end{aligned}$$

$$\begin{aligned}\mathit{eat} &:: [=\mathrm{D}, \mathrm{V}],\\ \mathbf{\Psi}(\mathit{eat}) &= \mathbf{eat},\\ \mathbf{\Phi}(\mathit{eat}) &= E_{lk} \in \mathsf{V} \otimes \mathsf{D}\end{aligned}$$

$$\begin{aligned}\varnothing_v &:: [=\mathrm{V}, =\mathrm{D}, \mathrm{v}],\\ \mathbf{\Psi}(\varnothing_v) &= \mathbf{o},\\ \mathbf{\Phi}(\varnothing_v) &= O_{njl} \in \mathsf{U} \otimes \mathsf{D} \otimes \mathsf{V}\end{aligned}$$

$$\begin{aligned}\mathit{Alice} &:: [\mathrm{D}, \mathrm{nom}^-],\\ \mathbf{\Psi}(\mathit{Alice}) &= \mathbf{Alice},\\ \mathbf{\Phi}(\mathit{Alice}) &= A_j \in \mathsf{D}\end{aligned}$$

$$\begin{aligned}\varnothing_T &:: [=\mathrm{v}, \mathrm{nom}^+, \mathrm{T}],\\ \mathbf{\Psi}(\varnothing_T) &= \mathbf{p},\\ \mathbf{\Phi}(\varnothing_T) &= P_{mn} \in \mathsf{T} \otimes \mathsf{U}\end{aligned}$$

$$\begin{aligned}\mathit{did} &:: [=\mathrm{T}, \mathrm{wh}^+, \mathrm{C}],\\ \mathbf{\Psi}(\mathit{did}) &= \mathbf{did},\\ \mathbf{\Phi}(\mathit{did}) &= D_{im} \in \mathsf{C} \otimes \mathsf{T}\end{aligned}$$

We now derive a representation for the expression *What did Alice eat* (see Figure 4).

Step 1. We first merge *eat* and *what*, checking the $= \mathrm{D}$ feature from *eat* and the D feature from *what*. This gives us a subtree for the expression *eat what*, with feature string $[\mathrm{V}]$. In tensor-space,

[2] U is the vector space for the category v (i.e. little v, Koopman and Sportiche, 1991), while $\varnothing_v$ and $\varnothing_T$ are the empty phonological exponents for the syntactic categories v and T, respectively.

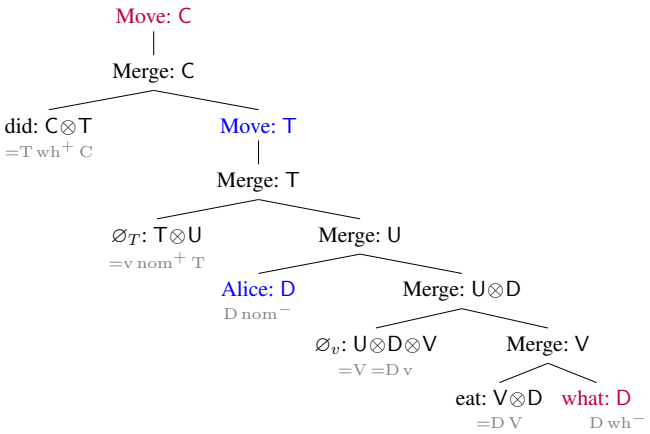


Figure 4: An MG derivation for "*What did Alice eat*", showing the semantic space for each lexical item and for each substructure as the result of merge and move operations.

syntactically we are **merge**-ing **eat** and **what**, and semantically we are contracting E_{lk} and W_k:

$$\mathcal{M}erge((\textbf{eat},E_{lk}),(\textbf{what},W_k)) = (\textbf{merge}(\textbf{eat},\textbf{what}),E_l)$$

with $\boldsymbol{\Phi}(\textit{eat what}) = E_{lk} \cdot W_k = E_l \in \mathsf{V}$, paralleling the remaining [V] feature of the substructure *eat what*. While *what* still has an active wh^- feature to allow for its displacement later in the derivation, recall that movement features are not reflected in the semantic space. For convenience, we write $t_0 := \textbf{merge}(\textbf{eat},\textbf{what})$.

Step 2. We now merge $\varnothing_v$ with *eat what*, checking the =V feature from $\varnothing_v$ and the V feature from *eat what*. The resulting expression $\varnothing_v$ *eat what* has feature string [=D, v]. In tensor-space:

$$\mathcal{M}erge((\textbf{o},O_{njl}),(t_0,E_l)) = (\textbf{merge}(\textbf{o},t_0),O_{nj})$$

Semantically, $\boldsymbol{\Phi}(\varnothing_v \textit{eat what}) = O_{njl} \cdot E_l = O_{nj} \in \mathsf{U}\otimes\mathsf{D}$, as expected. We let $t_1 := \textbf{merge}(\textbf{o},t_0)$.

Step 3. We merge $\varnothing_v$ *eat what* with *Alice*, checking the remaining =D feature from $\varnothing_v$ *eat what* and the D feature from *Alice*. This gives us *Alice* $\varnothing_v$ *eat what*, with feature string [v]. In vector space:

$$\mathcal{M}erge((t_1,O_{nj}),(\textbf{Alice},A_j)) = (\textbf{merge}(t_1,\textbf{Alice}),O_n)$$

with $O_{nj} \cdot A_j = O_n \in \mathsf{U}$, and $t_2 := \textbf{merge}(t_1,\textbf{Alice})$.

Step 4. The next merge gives us a structure for $\varnothing_T$ *Alice* $\varnothing_v$ *eat what*. For the vectors:

$$\mathcal{M}erge((\textbf{p},P_{mn}),(t_2,O_n)) = (\textbf{merge}(\textbf{p},t_2),P_m)$$

with $P_{mn} \cdot O_n = P_m \in \mathsf{T}$, and we let $t_3 := \textbf{merge}(\textbf{p},t_2)$.

Step 5. We encounter our first instance of move, triggered by the nom^+ feature introduced by $\varnothing_T$ checking the remaining nom^- feature on *Alice*. This results in the expression *Alice* $\varnothing_T$ $\varnothing_v$ *eat what*. For the vectors, syntactically we are applying the **move** operation to t_3. Recall that $\textbf{move} : \mathscr{F} \rightharpoonup \mathscr{F}$ is defined such that for any MG tree t, $\textbf{move}(\boldsymbol{\Psi}(t)) = \boldsymbol{\Psi}(\mathrm{move}(t))$. Semantically, there is no change in the vector space. That is:

$$\mathcal{M}ove((t_3,P_m)) = (\textbf{move}(t_3),P_m)$$

Again we let $t_4 := \textbf{move}(t_3)$.

Step 6. We merge *did* and *Alice* $\varnothing_T$ $\varnothing_v$ *eat what*, checking the =T feature from *did* and the T feature from *Alice* $\varnothing_T$ $\varnothing_v$ *eat what*. For the tensors:

$$\mathcal{M}erge((\textbf{did},D_{im}),(t_4,P_m)) = (\textbf{merge}(\textbf{did},t_4),D_i)$$

so that $D_{im} \cdot P_m = D_i \in \mathsf{C}$. We let $t_5 := \textbf{merge}(\textbf{did},t_4)$.

Step 7. Finally, one last move operation is triggered by the remaining wh^- feature on *what*, which is now checked against the active wh^+ feature introduced by *did*. This results in the expression *what did* $\varnothing_T$ *Alice* $\varnothing_v$ *eat*, and the only remaining feature associated with the tree structure is $[\mathrm{C}]$. In the vector space, we have $\mathbf{move}(t_5)$ for the syntactic representation: Semantically, there is again no change in the vector space.

$$\begin{aligned}&\mathcal{M}ove((t_5,D_i))\\=&(\mathbf{move}(t_5),D_i)\end{aligned}$$

Figure 4 illustrates this full derivation, focusing on how the tensor-semantics happens in parallel to the symbolic feature-checking.

6 Discussion and Conclusion

In this paper, we constructed a tensor-semantics for MGs, illustrating how distributional approaches to lexical semantics can be enhanced by the compositional operations of MGs. Specifically, we built on beim Graben and Gerth (2012)'s formulation of merge and move as tensor-product operations over structural positions, and we added maps to tensor-based semantics representations from MG syntactic features (Maillard et al., 2014). Notably, we restricted our construction to cases in which the lexical semantics of a mover is fully integrated into the derivation when said mover is first merged. This is consistent with an interpretation of move as "internal merge" not contributing any novel lexical semantics to the derivational process.

However, move is often conceived as interacting with propositional meaning in more complex ways, for instance by affecting quantifier scope taking and variable biding. In this sense, our construction is a tensor-based implementation of Kobele (2012)'s moveEmpty operation, which does not affect the denotational content of a lexical item. Kobele (2012) also allows move to postpone the incorporation of the semantics of a mover to one of its landing sites via a type of storage. This approach is compatible with our framework, and it would require to delay the tensor-contraction associated with the selection of a lexical item containing a movement feature. We conjecture that this extension is straightforward, since the mappings of these extra move operations are still homomorphisms between partial algebras. Future work should define the technical details of this approach, while also evaluating alternative formalizations to the effects of movement on propositional semantics across clause types (see also Kobele, 2006; Hunter, 2007; Tomita, 2016).

In line with Maillard et al. (2014)'s work on CCGs, here we made no assumptions about how the semantic tensors should be interpreted. While we tie our framework to distributional semantics broadly construed, future work should investigate how semantically interpretable vectors for the different types of features can be acquired (Bos et al., 2004; Socher et al., 2007, 2012; Hermann and Blunsom, 2013). Encouragingly, recent work on broad-coverage neural parsers for MGs has shown unexpected success in extracting from corpora super-tags for MG-style lexical items, even for the controversial "phonologically null" syntactic categories (Torr, 2017, 2018; Torr et al., 2019). These results leverage heavily work on broad-coverage CCG parsing via super-tagging (Bangalore and Joshi, 1999), and suggest that several enterprises (learning semantic tensors, super-tag inference, and broad-coverage syntactic and semantic parsing) can benefit from insights across grammar formalisms (Kasai et al., 2017; Prange et al., 2021; Yamaki et al., 2023).

Finally, the addition of tensor-based semantics to the geometric MG formalization of beim Graben and Gerth (2012) opens the way to future work incorporating semantic information into dynamical models of parsing, with potential relevance to cognitively plausible models of sentence processing (Gerth and beim Graben, 2009; Smith and Vasishth, 2020; Villata and Tabor, 2022; De Santo, 2025). Soulos (2025) showcases a general framework for integrating tensor representations of compositional structure with various connectionist accounts of learning and processing, in line with Smolensky et al. (2022) proposal for neurosymbolic computing. Overall then, our results add another piece to the puzzle of how to successfully connect generative syntax to broader work on neuro-symbolic processing in cognitive science (Smolensky et al., 1993; Smolensky, 2012; Martin, 2020; Garcez and Lamb, 2023)

References

David Adger. 2003. *Core syntax: A minimalist approach*. Oxford University Press.

Srinivas Bangalore and Aravind Joshi. 1999. Supertagging: An approach to almost parsing. *Computational linguistics*, 25(2):237–265.

Marco Baroni. 2013. Composition in distributional

semantics. *Language and Linguistics Compass*, 7(10):511–522.

Marco Baroni, Raffaella Bernardi, and Roberto Zamparelli. 2014. Frege in space: A program for composition distributional semantics. *Linguistic Issues in Language Technology*, 9.

Peter beim Graben and Sabrina Gerth. 2012. Geometric representations for minimalist grammars. *Journal of logic, language, and information*, 21(4):393–432.

Gemma Boleda. 2020. Distributional semantics and linguistic theory. *Annual Review of Linguistics*, 6:213–234.

Gemma Boleda and Aurélie Herbelot. 2016. Formal distributional semantics: Introduction to the special issue. *Computational Linguistics*, 42(4):619–635.

Johan Bos, Stephen Clark, Mark Steedman, James R Curran, and Julia Hockenmaier. 2004. Wide-coverage semantic representations from a ccg parser. In *COLING 2004: Proceedings of the 20th international conference on computational linguistics*, pages 1240–1246.

Stephen Clark. 2015. Vector space models of lexical meaning. *The Handbook of Contemporary semantic theory*, pages 493–522.

Stephen Clark and Stephen Pulman. 2007. Combining symbolic and distributional models of meaning. In *Proceedings of AAAI Spring Symposum on Quantum Interaction*.

Daoud Clarke. 2009. Context-theoretic semantics for natural language: an overview. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 112–119, Athens, Greece. Association for Computational Linguistics.

Daoud Clarke, David Weir, and Rudi Lutz. 2011. Algebraic approaches to compositional distributional semantics. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis*, 36:345–384.

Elizabeth Coppock and Lucas Champollion. 2025. Invitation to formal semantics. *Manuscript, Boston University and New York University. Last Updated October 2025*.

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.

Aniello De Santo. 2025. Capturing online SRC/ORC effort with memory measures from a minimalist parser. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 24–35, Albuquerque, New Mexico, USA. Association for Computational Linguistics.

Katrin Erk. 2012. Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653.

Katrin Erk. 2022. The probabilistic turn in semantics and pragmatics. *Annual Review of Linguistics*, 8:101–121.

James Fodor and Shinsuke Suzuki. 2026. Using computational semantics to study meaning in the brain. *Neuroscience & Biobehavioral Reviews*, 181:106514.

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.

Werner Frey and Hans-Martin Gärtner. On the treatment of scrambling and adjunction in minimalist grammars. In *Proceedings of formal grammar*, volume 2002, pages 41–52.

Artur d'Avila Garcez and Luis C Lamb. 2023. Neurosymbolic AI: The 3rd wave. *Artificial Intelligence Review*, 56(11):12387–12406.

Sabrina Gerth and Peter beim Graben. 2009. Unifying syntactic theory and sentence processing difficulty through a connectionist minimalist parser. *Cognitive neurodynamics*, 3(4):297–316.

Thomas Graf. 2022. Subregular linguistics: bridging theoretical linguistics and formal grammar. *Theoretical Linguistics*, 48(3-4):145–184.

Edward Grefenstette. 2013a. *Category-theoretic quantitative compositional distributional models of natural language semantics*. Ph.D. thesis, Oxford University, UK.

Edward Grefenstette. 2013b. Towards a formal distributional semantics: Simulating logical calculi with tensors. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 1–10, Atlanta, Georgia, USA. Association for Computational Linguistics.

Edward Grefenstette, Mehrnoosh Sadrzadeh, Stephen Clark, Bob Coecke, and Stephen Pulman. 2011. Concrete sentence spaces for compositional distributional models of meaning. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Irene Heim and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Blackwell, Malden, MA.

Karl Moritz Hermann and Phil Blunsom. 2013. The role of syntax in vector space models of compositional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 894–904, Sofia, Bulgaria. Association for Computational Linguistics.

Norbert Hornstein, Jairo Nunes, and Kleanthes K Grohmann. 2005. *Understanding minimalism*. Cambridge University Press.

Tim Hunter. 2007. Deriving syntactic properties of arguments and adjuncts from neo-davidsonian semantics. In *Conference on Mathematics of Language*, pages 103–116. Springer.

Shinnosuke Isono. 2024. Category locality theory: A unified account of locality effects in sentence comprehension. *Cognition*, 247:105766.

Jungo Kasai, Robert Frank, R. Thomas McCoy, Owen Rambow, and Alexis Nasr. 2017. TAG parsing with neural networks and vector representations of supertags. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1712–1722, Copenhagen, Denmark. Association for Computational Linguistics.

Gregory M. Kobele. 2006. *Generating copies: An investigation into structural identity in language and grammar*. Ph.D. thesis, University of California, Los Angeles.

Gregory M. Kobele. 2012. Importing montagovian dynamics into minimalism. In *Logical Aspects of Computational Linguistics - 7th International Conference, LACL 2012, Nantes, France, July 2-4, 2012. Proceedings*, Lecture Notes in Computer Science, pages 103–118. Springer.

Gregory M. Kobele, Sabrina Gerth, and John Hale. 2012. Memory resource allocation in top-down minimalist parsing. In *Formal Grammar - 17th and 18th International Conferences, FG 2012, Opole, Poland, August 2012, Revised Selected Papers, FG 2013, Düsseldorf, Germany, August 2013. Proceedings*, Lecture Notes in Computer Science, pages 32–51. Springer.

Tamara G Kolda and Brett W Bader. 2009. Tensor decompositions and applications. *SIAM review*, 51(3):455–500.

Hilda Koopman and Dominique Sportiche. 1991. The position of subjects. *Lingua*, 85(2-3):211–258.

Joachim Lambek. 1958. The mathematics of sentence structure. *The American Mathematical Monthly*, 65(3):189–224.

Serge Lang. 2002. The tensor product. In *Algebra*, chapter XVI, pages 601–640. Springer New York, New York, NY.

Alessandro Lenci. 2018. Distributional models of word meaning. *Annual review of Linguistics*, 4:151–171.

Jean Maillard, Stephen Clark, and Edward Grefenstette. 2014. A type-driven tensor-based semantics for CCG. In *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, pages 46–54, Gothenburg, Sweden. Association for Computational Linguistics.

Andrea E Martin. 2020. A compositional neural architecture for language. *Journal of Cognitive Neuroscience*, 32(8):1407–1427.

Tomáš 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.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio. Association for Computational Linguistics.

Richard Montague. 1970. English as a formal language. In B. Visentini et al, editor, *Linguaggi nella Società e nella Tecnica*, pages 188–221. Milan: Edizioni di Communita. Reprinted in Thomason (ed.) 1974, pp. 188–221.

Satoru Ozaki, Aniello De Santo, Tal Linzen, and Brian Dillon. 2024. CCG parsing effort and surprisal jointly predict RT but underpredict gardenpath effects. In *Society for Computation in Linguistics*, volume 7, pages 362–364.

Jakob Prange, Nathan Schneider, and Vivek Srikumar. 2021. Supertagging the long tail with tree-structured decoding of complex categories. *Transactions of the Association for Computational Linguistics*, 9:243–260.

Daniel Quigley. 2025. A vector logic for extensional formal semantics. *Journal of Logic, Language and Information*, pages 557–599.

Garrett Smith and Shravan Vasishth. 2020. A principled approach to feature selection in models of sentence processing. *Cognitive science*, 44(12):e12918.

Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial intelligence*, 46(1-2):159–216.

Paul Smolensky. 2012. Symbolic functions from neural computation. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1971):3543–3569.

Paul Smolensky and Géraldine Legendre. 2006. *The harmonic mind: From neural computation to optimality-theoretic grammar (Cognitive architecture), Vol. 1*. MIT Press.

Paul Smolensky, Géraldine Legendre, and Yoshiro Miyata. 1993. Integrating connectionist and symbolic computation for the theory of language. *Current Science*, pages 381–391.

Paul Smolensky, Richard McCoy, Roland Fernandez, Matthew Goldrick, and Jianfeng Gao. 2022. Neurocompositional computing: From the central paradox of cognition to a new generation of ai systems. *AI Magazine*, 43(3):308–322.

Richard Socher, Chris Biemann, and Rainer Osswald. 2007. Combining contexts in lexicon learning for semantic parsing. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*, pages 175–182, Tartu, Estonia. University of Tartu, Estonia.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211, Jeju Island, Korea. Association for Computational Linguistics.

Paul Soulos. 2025. *Unifying Neural and Symbolic Computation for Compositional Generalization: Representation and Processing*. Ph.D. thesis, The Johns Hopkins University.

Edward Stabler. 1996. Derivational minimalism. In *International conference on logical aspects of computational linguistics*, pages 68–95. Springer.

Edward P Stabler. 2013. Two models of minimalist, incremental syntactic analysis. *Topics in cognitive science*, 5(3):611–633.

Miloš Stanojević, Jonathan R Brennan, Donald Dunagan, Mark Steedman, and John T Hale. 2023. Modeling structure-building in the brain with ccg parsing and large language models. *Cognitive science*, 47(7):e13312.

Mark Steedman. 2001. *The syntactic process*. MIT press.

Mark Steedman and Jason Baldridge. 2011. Combinatory categorial grammar. In Robert D. Borsley and Kersti Börjars, editors, *Non-Transformational Syntax*, chapter 5, pages 181–224. John Wiley & Sons, Ltd.

Whitney Tabor. 2009. A dynamical systems perspective on the relationship between symbolic and non-symbolic computation. *Cognitive neurodynamics*, 3(4):415–427.

Yu Tomita. 2016. Solving event quantification and free variable problems in semantics for minimalist grammars. In *Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation: Oral Papers*, pages 219–227.

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.

John Torr. 2018. Constraining mgbank: Agreement, l-selection and supertagging in minimalist grammars. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 590–600.

John Torr, Miloš Stanojević, Mark Steedman, and Shay B Cohen. 2019. Wide-coverage neural a* parsing for minimalist grammars. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 2486–2505.

Sandra Villata and Whitney Tabor. 2022. A self-organized sentence processing theory of gradience: The case of islands. *Cognition*, 222:104943.

Ryosuke Yamaki, Tadahiro Taniguchi, and Daichi Mochihashi. 2023. Holographic ccg parsing. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 262–276.

Word Predictability on Code-switching Points in Cantonese–English Discourse

Ariel Chan
University of California, Santa Cruz
arielchan@ucsc.edu

Yanting Li
University of California, Irvine
yantil5@uci.edu

Jacob Poschl
University of California, Santa Cruz
jposchl@ucsc.edu

Abstract

This paper investigates how word predictability influences code-switching probability. We analyze 1,010 code-switched instances drawn from naturalistic sociolinguistic interviews with 41 Cantonese–English bilinguals across three bilingual groups (homeland, immersed, and heritage). In particular, we examine whether the predictability of switch points, operationalized as surprisal, influences the likelihood of code-switching. Using pretrained transformer-based language models, we estimate surprisal at the switch point under different modeling conditions, including autoregressive and masked models and varying amounts of contextual information. Mixed-effects logistic regression analyses show that less predictable words are more likely to be code-switched. These effects are largely consistent across model types and bilingual groups. Overall, these findings highlight the role of predictability in bilingual speech production and provide new insights into code-switching among bilingual speakers with diverse language experiences.

1 Introduction

Code-switching, the alternation between two or more languages within a conversation, is a prevalent phenomenon among bi/multilinguals (Milroy and Muysken, 1995; Poplack, 1988). It offers unique insights into the interaction between sociolinguistics, cognition, and language processing (Kroff and Dussias, 2023; Stell and Yakpo, 2015). Because of this, code-switching has attracted sustained attention from psycholinguistics, sociolinguistics, cognitive scientists, computational linguistics and more.

This paper zooms in onto intra-sentential code-switches, that is, switches that occur within a sentence, and investigates the factors shaping the choice of the switching point. Specifically, we examine the role of word predictability given context, which has been found in previous corpus studies and behavioral experiments to influence code-switching behaviors (Myslín and Levy, 2015; Calvillo et al., 2020; Bhattacharya and van Schijndel, 2026; de Bruin and Shiron, 2024). However, in prior work, predictability has typically been estimated based solely on the *preceding* context of the code-switched element, which does not fully capture the information available to the *speaker*. Given a communicative goal, speakers are likely to possess at least a partially specified representation of the upcoming message even if it is not fully lexicalized, which may also shape expectations about the current word (Bell et al., 2009; Pluymaekers et al., 2005; Upadhye and Futrell, 2025).

Another dimension missing from previous analyses is the language background of the speakers, which can strongly influence code-switching behaviors, including the frequency and type of switches (Poplack, 1980; Backus, 2003; Gollan and Ferreira, 2009; Kootstra et al., 2012; Blanco-Elorrieta and Caramazza, 2021), and potentially the switching point as well. For instance, less balanced bilinguals may have faster or more reliable lexical access in their dominant language, making words in that language more predictable overall. In contrast, balanced speakers with similar proficiency in both languages may experience greater uncertainty in lexical selection, which could influence the way predictability interacts with code-switching, although the direction and magnitude of this effect may vary. Despite these insights, few studies have systematically examined how word-level predictability interacts with speaker language background to shape intra-sentential code-switching.

Drawing on a naturalistic corpus of Cantonese–English code-switching, this study investigates the effect of word predictability on bilingual code-switching probability. The analysis focuses on single-word switches to English within Cantonese-majority sentences, comparing code-switched and

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 230–243
July 3-4, 2026 ©2026 Association for Computational Linguistics

non-code-switched words occurring in similar syntactic environments. Word probability is estimated using both autoregressive and bidirectional masked language models and with different context lengths to test whether code-switching is better predicted by models that incorporate information beyond the immediately preceding context. Additionally, to capture variation in bilingual language experience and sociocultural context, we examine three groups of bilingual speakers with distinct acquisition contexts and interactional contexts: homeland bilinguals, immersed bilinguals and heritage bilinguals. Specifically, this study addresses the following three research questions:

- Are code-switched words in Cantonese–English speech less predictable given context?
- Does the effect of predictability on code-switching change depending on how context is defined—for example, when using longer preceding context or when incorporating both preceding and following contexts?
- Does the effect of predictability on code-switching differ across bilingual groups with distinct acquisition and interactional contexts (homeland, immersed, and heritage)?

Because most studies examine a single bilingual population, it remains unclear whether predictability effects vary across bilingual groups with different language experiences. We do not expect a main effect of bilingual group; however, we predict an interaction between bilingual group and surprisal, such that the surprisal effect may differ across groups: stronger for homeland and immersed bilinguals, whose language use more closely aligns with the texts on which the LMs were trained, and smaller and potentially not significant for heritage bilinguals.

2 Background

Cantonese–English bilinguals provide a unique opportunity to examine code-switching, as their conversations often exhibit dynamic and complex switching patterns (Gibbons, 1987; Luke, 1998; Li, 2000). This is partially due to the historical development of Cantonese diasporic communities: Hong Kong has a history of being a British colony (Bolton, 2000; Sung, 2015), and there has been a significant migration of Cantonese speakers to the United States (Chang, 2004). Fig 1 shows an

咁 都 清楚 因為 譬如 藏族人 同 滿族人 就 assimilate 咗
That's clear because, for example, Tibetans and Manchus have assimilated.

Figure 1: Example of intra-sentential Cantonese–English code-switch.

example of a code-switched sentence where the majority part of the sentence is in Cantonese, with a single-word switch to English.

Why bilinguals code-switch has always been an interesting question. Various factors have been identified in the literature, including sociolinguistic considerations such as the identity of the speaker and listener (Chen, 2008; Lo, 1999), register (Halmari and Smith, 1994), indexicality (Myers-Scotton, 2020), discourse-functional purposes such as to emphasize important information (Gumperz, 1982; Myers-Scotton, 1993) and to strengthen discourse cohesiveness by maximizing the salience of discourse markers (De Rooij, 2000), as well as linguistic factors such as word length (Bhattacharya and van Schijndel, 2026; Calvillo et al., 2020; Myslín and Levy, 2015), syntactic roles (Myslín and Levy, 2015; Poplack, 1980), cognate priming (Broersma and De Bot, 2006; Kootstra et al., 2012), lexical gap filling (Grosjean, 1982; Li et al., 2024; Otheguy and García, 1993), and more.

Among these linguistic factors, word predictability has attracted significant attention in recent years, and it is also the main focus of this study. It has been found across languages that code-switched words tend to be less predictable given context. Using a corpus of naturalistic Czech–English bilingual conversation, Myslín and Levy (2015) designed a cloze test for sentence-final words that were either code-switched or not. Bilinguals listened to each sentence up to the final word and guessed the final word in either language. The results revealed that code-switched words were harder to guess, suggesting a lower predictability.

In another study, Calvillo et al. (2020) examined the written communication of Mandarin–English bilinguals and found the same effect. They estimated word probability by calculating surprisal, defined as the negative logarithmic probability of a word given its context (Hale, 2001, 2016; Levy, 2008), using a 5-gram model. Bhattacharya and van Schijndel (2026) replicated these findings using large language models (LLMs) and further extended them from written data to a speech corpus of the same language pair, providing cross-model evidence of robustness of the surprisal effect.

In addition to corpora studies, another piece of evidence comes from de Bruin and Shiron (2024)'s Bulgarian–English production experiment. In the experiment, participants were presented with a carrier sentence followed by a picture, and were instructed to read the sentence out loud and complete it by naming the picture in either language. The carrier sentences either made the pictures more predictable (e.g. *She packs the* + bag) or not (*They invite the* + singer). The results showed that code-switching was more likely to happen when the pictures were less predictable given the contexts.

Evidence from both experimental and computational studies have suggested a robust effect of predictability on code-switching. To account for this effect, Myslín and Levy (2015) proposed that this is to help alert the audience, which is supported by Bhattacharya and van Schijndel (2026). Knowing that an upcoming word is unpredictable, the speaker switches to another language, and such a salient language change will invite the listener to pay more attention to the content, making comprehension easier. Calvillo et al. (2020) on the other hand, offered explanations from a speaker-oriented angle: code-switching occurs at less predictable words because the increased cognitive demand to retrieve such words leaves fewer resources for inhibiting the non-target language, resulting in code-switching. de Bruin and Shiron (2024)'s explanation is also more along the line of this speaker-oriented proposal, as they found that predictive carrier sentences lowered code-switching frequency compared to no-context picture naming, but unpredictive carrier sentences did not necessarily increase code-switching frequency. The authors suggested that when contextual cues make an upcoming word more predictable, the co-activation of its translation equivalents in the other language is reduced, leading to fewer code-switches.

In all studies above, predictability is estimated using preceding context.[1] However, to understand why *speakers* choose to code-switch at a certain point, it is crucial to consider the future context in addition to the preceding context. Unlike comprehenders who process sentences incrementally, speakers have access to at least partial future message even if it is not fully planned, and the upcoming content may influence production at the current time point, from acoustic realization to word choice (Bell et al., 2009; Ranjan et al., 2022; Pluymaekers et al., 2005; Zhan and Levy, 2019; Upadhye and Futrell, 2025). In the context of code-switching, this forward-looking planning is particularly important: a speaker may anticipate that an upcoming word or syntactic structure is more easily accessed in one language than the other, or that a switch aligns better with the overall sentence plan. As a result, the decision to switch languages at a certain word can be shaped not only by the preceding context but also by the planned upcoming elements. Incorporating future context may therefore help better capture the planning processes underlying code-switching behavior.

[1] In de Bruin and Shiron (2024), the code-switching point always occurred at the end of a trial, so there was no following context. For the other studies (Myslín and Levy, 2015; Calvillo et al., 2020; Bhattacharya and van Schijndel, 2026), code-switches could be followed by additional content, either within the same sentence or within the same conversation.

3 Method

To answer our research questions, we need words that appear in comparable syntactic environments, with some instances involving a code-switch and others remaining unilingual. We then estimate and compare their predictabilities using pre-trained LMs. The procedures are described below.

3.1 Corpus

We use the CANGLISH Bilingual Corpus (Chan, 2023), which includes conversational speech from 41 Cantonese–English bilinguals across three bilingual groups: 13 homeland bilinguals raised in a Cantonese-dominant environment and educated in English in Hong Kong, 14 heritage bilinguals raised in Cantonese-speaking households in the U.S., and 14 immersed bilinguals from Hong Kong who later moved to the U.S. All participants acquired Cantonese as their first language and English as their second.

Data were collected online via Zoom between September and December, 2022, containing 36 hours and 17 minutes of audio and video recordings. Each participant completed two tasks: (1) a map task in which the participant and experimenter took turns describing slides to match each other's and (2) a sociolinguistic interview exploring language use and identity between the participant and experimenter. Code-switching was naturally introduced by the experimenter with language switches approximately every 10 minutes. Participants were informed that they could use either Cantonese, English, or both during the conversation.

3.2 Preprocessing

The recorded interviews were transcribed and annotated using the ELAN software (Version 6.8) (Wittenburg et al., 2006) manually by highly proficient Cantonese–English bilinguals, with sensitive or identifying information masked. Transcriptions were extracted from 41 ELAN `.eaf` files using the `pympi` (Lubbers and Torreira, 2013-2025) library. Sentence boundaries were taken directly from human annotations, and each annotated span was treated as a sentence without further automatic segmentation. After sentence segmentation, a cleaning pipeilne was applied that removed anonymization/annotation markers, normalized whitespace, and removed punctuation tokens and filler words (e.g., eh, oh, 呃, 嗯, see Appendix A for a full list).

Sentences containing fewer than two words after filtering were excluded. Cantonese and English tokens were identified using Unicode script detection (CJK vs. ASCII alphabetic characters). Cantonese text was tokenized with `pycantonese` (Lee et al., 2022) while English text was tokenized using whitespace segmentation. Full preprocessing/tokenization details are provided in Appendix A.

Sentences containing both Cantonese and English tokens were labeled as code-switched. A code-switched sentence's majority language was determined by word counts. Sentences with an equal word count for English and Cantonese were labeled "Equal". For the present analysis, we restricted our dataset to code-switched sentences with Cantonese as the majority language because switches from Cantonese to English represent the unmarked and most common switch direction both in our corpus and among Cantonese–English bilingual communities (Chan et al., 1983). We also made sure that at least two Cantonese words preceded the switching point to avoid sentence-initial switches and short fragments where predictability estimates were unreliable. One heritage participant did not produce any code-switched sentences that met these criteria and was therefore not included in the analysis.

3.3 Identifying Syntactically Matched Sentence Pairs

For each code-switched sentence, we translated the English span(s) into Cantonese using Meta's NLLB model (Costa-Jussà et al., 2022), and replaced the English span(s) with their Cantonese translations, producing a Cantonese-only sentence (see Appendix Table 3 for more deteails). The resulting Cantonese sentences were then POS-tagged using PyCantonese (Lee et al., 2022).

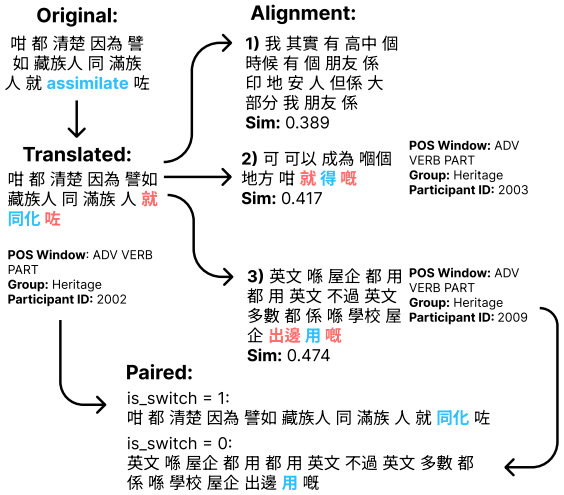


Figure 2: Demonstration of the matching procedure for a code-switched instance. The original sentence is translated into Cantonese and POS-tagged. Candidate 1 fails the similarity criterion; Candidates 2 and 3 pass and are ranked by match preference, yielding Candidate 3 as the selected match. The switch point (blue) is the index for surprisal calculation.

Figure 2 illustrates the procedure for identifying a syntactically similar unilingual Cantonese sentence. Following Calvillo et al. (2020), candidate sentences were required to satisfy two criteria: (i) the unilingual sentence achieved a Levenshtein similarity ≥ 0.4 with the translated code-switched sentence and (ii) the unilingual sentence contained the same POS-tag window of size three centered on the switch point (truncated at sentence boundaries). When multiple candidates satisfied these constraints, we selected a match using a deterministic ranking procedure. Same-speaker matches were preferred over matches from other speakers within the same bilingual group to control for speaker-specific lexical preferences and habitual code-switching patterns. When multiple same-speaker candidates were available, we selected the sentence closest in time to the code-switched sentence, under the assumption that temporally proximate utterances are more likely to share discourse topic and contextual constraints. Remaining ties were resolved by selecting the candidate with the highest similarity score. Using this procedure, 1,010 of the 1,250 code-switched sentences were successfully paired with a unilingual Cantonese

	Autoregressive LM	Masked LM
Code-switched sentence	$p(\text{同化} \mid c_{\text{pre}})$	$p(\text{同化} \mid c_{\text{pre}}, \texttt{[MASK]}, c_{\text{post}})$
Matching sentence	$p(\text{用} \mid c_{\text{pre}})$	$p(\text{用} \mid c_{\text{pre}}, \texttt{[MASK]}, c_{\text{post}})$

Table 1: Target words and contexts used for surprisal estimation without discourse context, illustrated using the example in Fig 2. c_{pre} and c_{post} denote the preceding and following context relative to the target word. The autoregressive model conditions only on c_{pre}; the masked model additionally incorporates c_{post} via masked prediction.

sentence, with a mean Levenshtein similarity of 0.464 between pairs (SD = 0.077).

3.4 Language Model Surprisal Estimation

Using our example in Fig 2, in this step, we calculate $p(\text{同化}|context)$ for the code-switched sentence, and $p(\text{用}|context)$ for the matching sentence for predicting whether a switch would happen. Next, we introduce the LMs used, as well as how contexts are defined for each model.

The two pretrained transformer LMs used are:

- `hon9kon9ize/bert-large-cantonese` (hon9kon9ize Group, 2024a): a masked LM
- `hon9kon9ize/CantoneseLLMChat-v1.0-7B` (hon9kon9ize Group, 2024b): an autoregressive LM

Table 1 illustrates the target word and contexts used for each model. For each sentence, both models were conditioned on the discourse context preceding the switch point. The masked model additionally received available post-switch content (i.e., words following the switch point) to allow bidirectional conditioning, whereas the autoregressive model only used the preceding context by design. Token-level probabilities at the first code-switched word were used to compute surprisal.

Let $w = w_{1:K}$ denote the target word and c the discourse context. Token surprisal is defined as:

$$S_k = -\log_2 P(w_k \mid c, k) \tag{1}$$

Word-level surprisal is the sum of token surprisals:

$$S(w \mid c) = \sum_{k=1}^{K} S_k \tag{2}$$

Token probabilities $P(\cdot \mid c, k)$ were obtained using left-to-right conditionals for the autoregressive model and a masking-based pseudo-log-likelihood procedure for the masked model (see Appendix C).

We also examined two context conditions: no preceding discourse context ($n = 0$, sentence-internal context only) and three preceding sentences ($n = 3$). In the $n = 3$ condition, any English spans in the preceding sentences were translated to Cantonese using the same NLLB procedure described in Section 3.3, ensuring that discourse context remained in-distribution for the Cantonese LMs. Instances where translation quality was insufficient were excluded (see Appendix C).

3.5 Regression

We tested whether LM-estimated surprisal at the switch point predicted code-switching using a generalized linear mixed-effects logistic regression model. All analyses were conducted in R (R Core Team, 2024) using the `lme4` package (Bates et al., 2015) within RStudio (Posit team, 2025). Each observation corresponded to the word at the potential switch point, which is either the translation of the English word in the code-switched sentence or the matched word in the unilingual matching sentence (e.g. 同化 or 用 in the example in Fig 2).

The dependent variable for our mixed-effect logistic regression models was binary: `is_switch` $= 1$ for code-switched instances and `is_switch` $= 0$ for matched unilingual instances. Predictor variables included word length, part of speech (POS), surprisal, bilingual group (homeland, immersed, heritage), and the interaction between surprisal and bilingual group. Word length was calculated as the number of Chinese characters in the segmented Cantonese target word.

$$\begin{aligned}\eta_{ij} &= \beta_0 + \beta_1 \text{WordLength}_{ij} + \beta_2 \text{POS}_{ij} \\ &\quad + \beta_3 \text{BilingualGroup}_{ij} + \beta_4 \text{Surprisal}_{ij} \\ &\quad + \beta_5 \text{BilingualGroup}_{ij}\text{Surprisal}_{ij} + u_{0j}\end{aligned} \tag{3}$$

where $\eta_{ij} = \text{logit}\big(P(\text{is_switch}_{ij} = 1)\big)$ and $u_{0j} \sim \mathcal{N}(0, \sigma_u^2)$.

We fit four models: (1) two models using sur-

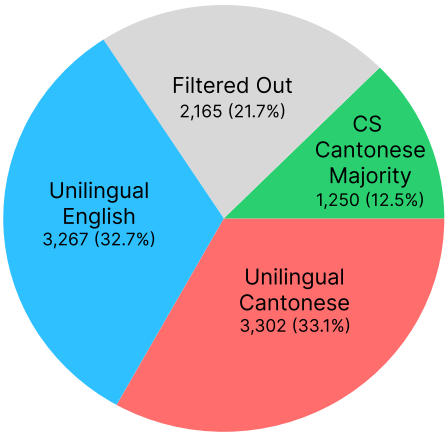


Figure 3: Distribution of sentences by language composition from a total of 9,984 initially processed sentences. The filtered-out section (grey) includes filler-only sentences (n=616), sentences below the minimum word threshold (n=990), code-switched sentences where English or neither language was the majority (n=230), and code-switched Cantonese-majority sentences excluded by the pattern filter or failed translation (n=329).

prisal estimated by the autoregressive LM with (1a) no preceding discourse context and (1b) three preceding sentences in the same discourse, and (2) two models using surprisal estimated by the masked LM with (2a) no preceding discourse context and (2b) three preceding sentences in the same discourse.

POS tags were collapsed into three categories: noun, verb, other.[2] Proper nouns were excluded, and analyses were restricted to single-worded switches. Categorical predictors (bilingual group and POS) were sum-coded using sum-to-zero contrasts, so coefficients represent deviations from the grand mean. Models also included a random intercept for participant to allow baseline switching probabilities to vary across speakers. Full model formulas are provided in Eq. 3.5.

4 Results

4.1 Descriptive Summary

The final dataset comprised 1,010 matched sentence pairs drawn from 1,250 code-switched sentences (Figure 3). Among these, 40.2% were contributed by the homeland group, 31.5% by the immersed group, and 28.3% by the heritage group. The two regression models without preceding discourse context (1a, 2a) each included 1,348 observations, whereas the two with three sentences of preceding context (1b, 2b) each included 1,346 observations, since two observations were at the beginning of the interview (i.e. without any preceiding discourse contexts).

4.2 Overall Effect of Predictability on Code-Switching

As specified above, we fit four models that varied in the type of language model (autoregressive vs. masked) and the amount of preceding discourse context (no prior discourse vs. three preceding sentences). Table 2 summarizes the results of all four models, including the direction of effect and significance level of each predictor. Detailed results for each model are reported in Appendix D. Across Models (1a), (2a), and (2b), surprisal showed a significant positive main effect, such that higher surprisal values were associated with a greater likelihood of code-switching. However in Model (1b), surprisal showed an opposite effect. Overall, the positive relationship between surprisal and code-switching is robust across most models, suggesting that lower word predictability generally increases the likelihood of switching.

To directly compare the predictive contributions of autoregressive and masked surprisal, we fit models containing autoregressive surprisal only, masked surprisal only, and both predictors simultaneously using three preceding sentences as discourse context. Likelihood ratio tests showed that adding masked surprisal to the autoregressive model significantly improved model fit, $\chi^2(3) = 33.62$, $p < .001$. In contrast, adding autoregressive surprisal to the masked model resulted in only a marginal improvement in fit, $\chi^2(3) = 7.43$, $p = .059$. Comparisons of model fit indices further supported this pattern: the masked-only model (AIC = 1703.27, BIC = 1755.32) substantially outperformed the autoregressive-only model (AIC = 1729.46, BIC = 1781.50), whereas the full model showed only a minimal improvement in AIC (1701.84) and a higher BIC (1769.50) due to increased model complexity. Together, these findings suggest that masked surprisal captures substantially more variance than autoregressive surprisal, although autoregressive surprisal may still contribute limited additional information when both measures are considered simultaneously.

[2] Tokens with unavailable or unclassifiable POS tags (e.g., returned as UNK or missing by the tagger) were excluded prior to regression analyses.

Model Type	**Autoregressive** (1a)	**Autoregressive** (1b)	**Masked** (2a)	**Masked** (2b)
Model Context	Sentence-internal	Sentence-internal + 3 Preceding Sentences	Sentence-internal	Sentence-internal + 3 Preceding Sentences
Word Length	$+^{***}$	$+^{***}$	$+^{***}$	$+^{***}$
POS: Noun	$-^{**}$	$-^{*}$	$-^{**}$	$-^{*}$
POS: Verb	*n.s.*	*n.s.*	*n.s.*	*n.s.*
Surprisal	$+^{***}$	$-^{**}$	$+^{***}$	$+^{***}$
Bilingual Group: Heritage	*n.s.*	*n.s.*	*n.s.*	*n.s.*
Bilingual Group: Immersed	*n.s.*	*n.s.*	*n.s.*	*n.s.*
Surprisal × Heritage	*n.s.*	*n.s.*	*n.s.*	*n.s.*
Surprisal × Immersed	*n.s.*	*n.s.*	*n.s.*	*n.s.*

Table 2: Summary of predictor significance across the four mixed-effects logistic regression models. Signs (+/−) indicate the direction of the log-odds estimate. Significance levels are marked as $*p < .05$, $**p < .01$, and $***p < .001$. Not-significant is labeled as *n.s.*.

4.3 Effects of Word Length, POS, and Bilingual Group

The main effects of word length, POS, and bilingual group were consistent across all four models. For ease of discussion, we report results from Model (2b). The interaction between surprisal and the heritage bilingual group showed a marginal trend (β = 0.03, SE = 0.02, z = 1.75, p = .080), suggesting a potential difference in the strength of the surprisal effect across groups, although this effect did not reach statistical significance. No significant interaction was observed for the immersed bilingual group (β = -0.01, SE = 0.01, z = -0.55, p = .583). There was also no significant main effect of bilingual group: neither heritage speakers (β = -0.19, SE = 0.20, z =-0.96, p = .339) nor immersed speakers (β = -0.02, SE = 0.19, z = 0.08, p = .937) differed significantly from the grand mean switching probability.

Word length had a significant positive effect on switching (β = 0.98, SE = 0.10, z = 9.58, p < .001), indicating that longer words were more likely to be switched. POS also influenced switching: the contrast for nouns was negative and significant (β = -0.21, SE = 0.08, z =-2.47, p = .013), indicating that nouns were less likely than the overall POS mean to be switched. The contrast for verbs was not significant (β = 0.06, SE = 0.10, z =0.64, p = .520).

5 Discussion

Previous work has shown that less predictable words are more likely to be code-switched, and this study is, to our knowledge, the first to extend this line of investigation to Cantonese–English bilingual speech, a relatively under-resourced language pair. In addition, we explored whether the predictability effect remained consistent across different types of LMs, different amounts of contextual information, and different bilingual groups. This comparison is novel and exploratory, as many previous studies focus on a single bilingual population from a relatively homogenous language context.

5.1 Evidence for a Predictability Effect in Cantonese–English Code-Switching

We first replicated the analysis of Calvillo et al. (2020) with the Cantonese–English dataset. Sentences containing a single-word switch from Cantonese to English were matched with unilingual Cantonese sentences with similar syntactic structure. After translating the switched word into Cantonese, we used an autoregressive model to estimate the word predictability at the switch (or matching) point given sentence-internal preceding context, and tested whether it predicted code-switching (Modal (1a)). The result was consistent with previous findings from other language pairs: less predictable words were more likely to be code-switched (Myslín and Levy, 2015; Calvillo et al., 2020; Bhattacharya and van Schijndel, 2026; de Bruin and Shiron, 2024). Additionally, we observed a word length effect consistent with the literature, with longer words being more likely to be code-switched (Myslín and Levy, 2015; Calvillo et al., 2020; Bhattacharya and van Schijndel, 2026). Meanwhile, nouns were less likely to be switched, which is a pattern that contrasts with the common prediction in the literature (Kroff, 2016; Chan, 1999), but it is consistent with the findings of Calvillo et al.. One possible explanation is that code-switched nouns can introduce new topics or discourse-salient referents (Barredo, 1997). Such

nouns are therefore typically less predictable. Because predictability was included as a predictor in our model, it may have captured some of the variance that would otherwise be attributed to part of speech, thereby reducing the independent effect of POS.

5.2 Robustness of the Predictability Effect across Language Models and Context Windows

To test the robustness of the predictability effect, we estimated surprisals using two types of models: an autoregressive model, which relies solely on preceding context, and a masked model, which allows bidirectional prediction on both preceding and following context. We also varied the amount of discourse information included for each model. When only the current sentence was used for surprisal estimation, the predictability effect was consistent across both models. However, when the preceding context was expanded to include the preceding three sentences in the same conversation, the effect reversed for the autoregressive model while remaining unchanged for the masked model.

Examining the correlation between surprisal estimates from no prior vs. extended contexts, we noticed a Pearson correlation of 0.78 for the masked model but only 0.11 for the autoregressive model, which may help explain the reversal in the surprisal effect. The low correlation for the autoregressive model may be due to the fact that all English words in the preceding context had to be translated to Cantonese, as the model was trained exclusively on Cantonese text. These translations may have introduced some unnaturalness that affected the surprisal estimates. In contrast, the masked model had access to both preceding and following context, which likely stabilized its predictions.

When autoregressive and masked surprisal were entered into the same model, masked surprisal remained a significant positive predictor of code-switching, while autoregressive surprisal exhibited a weaker negative association after controlling for masked surprisal. Likelihood ratio tests showed a consistent result: adding masked surprisal to the autoregressive model significantly improved model fit, while adding autoregressive surprisal to the masked model resulted only in a marginal improvement. Comparisons of AIC and BIC values similarly indicated that the masked model estimation substantially outperformed the autoregressive model estimation. Together, these findings suggest that the bidirectional masked language model provided a better prediction for code-switching than the autoregessive model that rely only on the preceding context. This result is consistent with the idea that bilingual speakers may make word choice decisions not only based on incremental context information, but also on expectations about the future, which are better captured by the bidirectional model. At the same time, the marginal contribution of autoregressive surprisal suggests that sequential prediction may still capture partially distinct aspects of bilingual predictive processing during code-switching.

5.3 Absence of a Bilingual Group Effect on Code-Switching Probability

Consistent with our expectation, we found no significant main effect of bilingual group. Contrary to our prediction, however, we also found no interaction between bilingual group and surprisal. This suggests the effect of predictability on switching probability were similar across heritage, immersed, and homeland bilinguals in our dataset. Although somewhat surprising, this pattern is consistent with findings from another study on comprehension of code-switched words in sentences (Chan, 2023). Together, these results suggest that bilingual speakers with diverse language acquisition backgrounds and interactional contexts may nevertheless exhibit similar patterns of code-switching. This provides a different perspective from studies that emphasize the role of bilinguals' language dominance or proficiency in shaping code-switching behavior (for a review, see Declerck and Philipp 2015).

One possible explanation is that cognitive mechanisms underlying lexical predictability and language selection operate similarly across bilingual speakers despite differences in interactional contexts. From a language control perspective (Green and Wei, 2014), speakers across groups may be similarly sensitive to predictability pressures when deciding whether to perform insertional lexical switches during language production. Alternatively, bilingual experience may primarily influence other aspects of code-switching (Gollan and Goldrick, 2018), such as language preference or discourse functions, rather than probability of switching at a particular lexical position. Our analysis also focused only on unidirectional lexical switches from Cantonese to English, the unmarked and default direction of switching in our corpus, which may further reduce the likelihood of observing an

interaction between bilingual group and surprisal. Finally, it is also possible that the absence of group effect reflects a selection bias: heritage bilinguals who agreed to participate in the study may represent a subgroup with relatively high self-rated Cantonese proficiency. It is therefore not surprising that they exhibited code-switching patterns similar to those of the other two groups.

6 Conclusion and Future Directions

This study provides evidence that word predictability plays a role in shaping intra-sentential single-word code-switches. Our findings indicated that less predictable words were more likely to be switched. This effect was largely consistent across different LMs, context windows, and bilingual groups, suggesting that predictability constraints operate similarly across speakers with diverse interactional contexts. However, several limitations should be noted. Our analysis focused on unidirectional lexical switches from Cantonese to English and did not consider additional switches in the surrounding discourse. Future work could examine English to Cantonese switches to see whether predictability effects generalize across switch directions.

Acknowledgments

We thank the research assistants in the Bilingualism, Mind, and Brain Lab at the University of California, Irvine, and the Bilingualism, Sociolinguistics, and Cognition (BiSoN) Lab at Stanford University and University of California, Santa Cruz for their assistance with this project. We also thank the reviewers and attendees of the 2024 California Meeting on Psycholinguistics (CAMP), the 38th Annual Conference on Human Sentence Processing (HSP), and Marten van Schijndel for their feedback and discussions. Data collection for this project was supported by a UCLA Dissertation Year Fellowship, a National Science Foundation Linguistics DDRI Grant (Award No. BCS-20424), and a Mellon/ACLS Dissertation Completion Fellowship awarded to Ariel Chan. This research was additionally supported by start-up funding from the Humanities Division at the University of California, Santa Cruz awarded to Ariel Chan, Assistant Professor of Languages and Applied Linguistics and Principal Investigator of the BiSoN Lab.

References

Ad Backus. 2003. Units in code switching: Evidence for multimorphemic elements in the lexicon. *Linguistics*, 41(1).

Inma Munoa Barredo. 1997. Pragmatic functions of code-switching among Basque–Spanish bilinguals. In *Proceedings of Actas do i Simposio Internacional sobre o Bilingismo*, pages 528–541.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67:1–48.

Alan Bell, Jason M Brenier, Michelle Gregory, Cynthia Girand, and Dan Jurafsky. 2009. Predictability effects on durations of content and function words in conversational English. *Journal of Memory and Language*, 60(1):92–111.

Debasmita Bhattacharya and Marten van Schijndel. 2026. Code-switching in text and speech challenges information-theoretic speaker design. *Language and Cognition*, 18:e31.

Esti Blanco-Elorrieta and Alfonso Caramazza. 2021. A common selection mechanism at each linguistic level in bilingual and monolingual language production. *Cognition*, 213:104625.

Kingsley Bolton. 2000. The sociolinguistics of Hong Kong and the space for Hong Kong English. *World Englishes*, 19(3):265–285.

Mirjam Broersma and Kees De Bot. 2006. Triggered codeswitching: A corpus-based evaluation of the original triggering hypothesis and a new alternative. *Bilingualism: Language and Cognition*, 9(1):1–13.

Jesús Calvillo, Le Fang, Jeremy Cole, and David Reitter. 2020. Surprisal predicts code-switching in Chinese-English bilingual text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4029–4039.

Ariel Shuk Ling Chan. 2023. *The Diaspora of Bilinguals: Code-Switching in Three Groups of Cantonese-English Bilinguals*. University of California, Los Angeles.

Brian Hok-Shing Chan. 1999. *Aspects of the syntax, production and pragmatics of code-switching-with special reference to Cantonese–English*. University of London, University College London (United Kingdom).

Mun-Chee Chan, Helen LH Chau, and Rumjahn Hoosain. 1983. Input/output switch in bilingual code switching. *Journal of Psycholinguistic Research*, 12(4):407–416.

Iris Chang. 2004. *The Chinese in America: A narrative history*. Penguin.

Katherine Chen. 2008. Positioning and repositioning: Linguistic practices and identity negotiation of overseas returning bilinguals in Hong Kong. *Multilingua*, 27(1/2):57.

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

Angela de Bruin and Veniamin Shiron. 2024. Putting language switching in context: Effects of sentence context and interlocutors on bilingual switching. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 50(7):1112.

Vincent A De Rooij. 2000. French discourse markers in Shaba Swahili conversations. *International Journal of Bilingualism*, 4(4):447–466.

Mathieu Declerck and Andrea M Philipp. 2015. A review of control processes and their locus in language switching. *Psychonomic Bulletin & Review*, 22(6):1630–1645.

John Gibbons. 1987. *Code-mixing and code choice: A Hong Kong case study*. Multilingual Matters.

Tamar H Gollan and Victor S Ferreira. 2009. Should I stay or should I switch? A cost–benefit analysis of voluntary language switching in young and aging bilinguals. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 35(3):640.

Tamar H Gollan and Matthew Goldrick. 2018. A switch is not a switch: Syntactically-driven bilingual language control. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 44(1):143.

David W Green and Li Wei. 2014. A control process model of code-switching. *Language, Cognition and Neuroscience*, 29(4):499–511.

François Grosjean. 1982. Code-switching and borrowing. In *Bilingual: Life and Reality*, pages 51–62. Harvard University Press.

John J. Gumperz. 1982. *Discourse Strategies*. Cambridge University Press, Cambridge, UK.

J. Hale. 2001. A probabilistic earley parser as a psycholinguistic model. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1–8. Association for Computational Linguistics.

John Hale. 2016. Information-theoretical complexity metrics. *Language and Linguistics Compass*, 10(9):397–412.

Helena Halmari and Wendy Smith. 1994. Code-switching and register shift: Evidence from Finnish-English child bilingual conversation. *Journal of Pragmatics*, 21(4):427–445.

hon9kon9ize Group. 2024a. bert-large-cantonese. https://huggingface.co/hon9kon9ize/bert-large-cantonese. Accessed: 2026-03-02.

hon9kon9ize Group. 2024b. Cantonesellmchat-v1.0-7b. https://huggingface.co/hon9kon9ize/CantoneseLLMChat-v1.0-7B. Accessed: 2026-03-02.

Gerrit Jan Kootstra, Janet G Van Hell, and Ton Dijkstra. 2012. Priming of code-switches in sentences: The role of lexical repetition, cognates, and language proficiency. *Bilingualism: Language and Cognition*, 15(4):797–819.

Jorge R Valdés Kroff. 2016. Mixed NPs in Spanish-English bilingual speech: Using a corpus-based approach to inform models of sentence processing. *Spanish-English Codeswitching in the Caribbean and the US*, 11:281.

Jorge R Valdés Kroff and Paola E Dussias. 2023. Production, processing, and prediction in bilingual codeswitching. In *Psychology of Learning and Motivation*, volume 78, pages 195–237. Elsevier.

Jackson Lee, Litong Chen, Charles Lam, Chaak Ming Lau, and Tsz-Him Tsui. 2022. PyCantonese: Cantonese linguistics and NLP in python. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6607–6611.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

David CS Li. 2000. Cantonese-English code-switching research in Hong Kong: A Y2K review. *World Englishes*, 19(3):305–322.

Yanting Li, Gregory Scontras, and Richard Rutrell. 2024. On the communicative utility of code-switching. In *Proceedings of the Society for Computation in Linguistics 2024*, pages 343–349.

Adrienne Lo. 1999. Codeswitching, speech community membership, and the construction of ethnic identity. *Journal of Sociolinguistics*, 3(4):461–479.

Mart Lubbers and Francisco Torreira. 2013-2025. pympi-ling: a Python module for processing ELANs EAF and Praats TextGrid annotation files. https://pypi.python.org/pypi/pympi-ling. Version 1.71.

Kang-Kwong Luke. 1998. Why two languages might be better than one: Motivations of language mixing in Hong Kong. In Martha C. Pennington, editor, *Language in Hong Kong at Century's End*, pages 145–159. Hong Kong University Press Hong Kong.

Lesley Milroy and Pieter Muysken. 1995. *One speaker, two languages: Cross-disciplinary perspectives on code-switching*. Cambridge University Press.

Carol Myers-Scotton. 1993. *Social motivations for codeswitching: Evidence from Africa*. Oxford University Press.

Carol Myers-Scotton. 2020. Code-switching as indexical of social negotiations. In *The Bilingualism Reader*, pages 97–122. Routledge.

Mark Myslín and Roger Levy. 2015. Code-switching and predictability of meaning in discourse. *Language*, pages 871–905.

Ricardo Otheguy and Ofelia García. 1993. Convergent conceptualizations as predictors of degree of contact in us spanish. *Spanish in the United States: Linguistic contact and diversity*, pages 135–154.

Mark Pluymaekers, Mirjam Ernestus, and R Baayen. 2005. Articulatory planning is continuous and sensitive to informational redundancy. *Phonetica*, 62(2-4):146–159.

Shana Poplack. 1980. Sometimes I'll start a sentence in Spanish Y TERMINO EN ESPAÑOL: toward a typology of code-switching. *Linguistics*, 18:581–618.

Shana Poplack. 1988. Contrasting patterns of code-switching in two communities. In Monica Heller, editor, *Codeswitching : Anthropological and Sociolinguistic Perspectives*, pages 215–244. De Gruyter, Inc.

Posit team. 2025. *RStudio: Integrated Development Environment for R*. Posit Software, PBC, Boston, MA.

R Core Team. 2024. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Sidharth Ranjan, Rajakrishnan Rajkumar, and Sumeet Agarwal. 2022. Linguistic complexity and planning effects on word duration in hindi read aloud speech. In *Proceedings of the Society for Computation in Linguistics 2022*, pages 119–132.

Gerald Stell and Kofi Yakpo, editors. 2015. *Code-switching between structural and sociolinguistic perspectives*. De Gruyter.

Chit Cheung Matthew Sung. 2015. Hong Kong English: linguistic and sociolinguistic perspectives. *Language and Linguistics Compass*, 9(6):256–270.

Shiva Upadhye and Richard Futrell. 2025. Back to the future: The role of past and future context predictability in incremental language production. *arXiv preprint arXiv:2511.07752*.

Peter Wittenburg, Hennie Brugman, Albert Russel, Alex Klassmann, and Han Sloetjes. 2006. ELAN: a professional framework for multimodality research. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Meilin Zhan and Roger Levy. 2019. Availability-based production predicts speakers' real-time choices of mandarin classifiers. *arXiv preprint arXiv:1905.07321*.

A Preprocessing Details

A.1 Implementation:

Raw transcriptions were stored in ELAN Annotation Format (EAF) files, one per participant. Each file was loaded using the pympi library (Lubbers and Torreira, 2013-2025), in which we can extract the relevant per-tier annotation data as `(start_time, end_time, text)` tuples. The participant tier was identified by its name prefix (ACHE, ACI, or ACH), and the interviewer tier (included for discourse context) was identified by the tier title: IR. Speaker group membership was assigned based on tier name prefix:

- ACHE → Heritage
- ACI → Immersed
- ACH → Homeland

A.2 Tokenization

Cantonese and English tokens were identified using Unicode script detection. Each annotation was first segmented into contiguous script runs: characters in CJK Unicode ranges were labeled Cantonese and ASCII alphabetic characters were labeled English. Cantonese segments were then tokenized using `pycantonese` (Lee et al., 2022) and English segments were split on whitespace. Some examples of segmentation:

- 我去*school*啦→ 我去(C), school (E), 啦(C)
- 我*local*人→ 我(C), local (E), 人(C)

Tokens were then cleaned by: removing anonymization markers (strings matching `^x+$`), stripping leading and trailing punctuation, normalizing Unicode dashes to ASCII hyphens and splitting on them, splitting on commas and ellipses, and discarding any tokens that were punctuation-only or empty after cleaning.

Filler and hesitation tokens were excluded prior to all analyses. Because matrix language determination relies on word counts in each language, retaining fillers would introduce noise due to the complexity of assigning such fillers to a particular language (e.g., *eh*). Filler detection was applied cross-lingually, so that a hesitation token was excluded regardless of which span it appeared in. The list of excluded tokens are as displayed in Figure 4.

Language	Fillers
English	uh, um, uhm, er, err, ah, eh, mm, hm, hmm, mhm, ehm, uh-huh, mm-hmm, uh-uh, mm-mm, em, emm, ehh, umm, ummm, uhh, mmm, huh, oh, ohh, ohhh, ummmm, uhmm
Cantonese	呃, 嗯, 啊, 哦, 唔, 吖, 哎, 咦

Figure 4: Filler and hesitation tokens excluded prior to all analyses, listed by language.

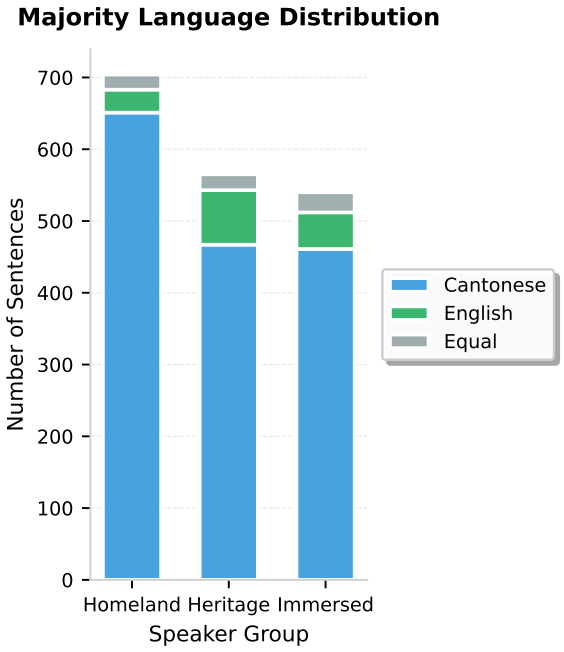


Figure 5: Distribution of sentences by majority language across the three bilingual speaker groups (homeland, heritage, and immersed).

B Sentence Matching and Alignment Examples

B.1 Translation and Truncation

Table 3 provides a concrete example of the transformation from a CS sentence to the counterfactual unilingual baseline used for surprisal scoring. NLLB occasionally fails to translate embedded English segments fully, leaving English tokens or malformed outputs. Rather than discarding these sentences entirely, we applied a targeted truncation procedure. After translation, each output was checked for residual English tokens. If any appeared before the switch point, the sentence was rejected outright. If they appeared only after the switch point (in or beyond the originally embedded English segment) the translation was truncated at the first offending token, provided the switch point itself was preserved. Sentences where truncation would have removed the switch point were also rejected.

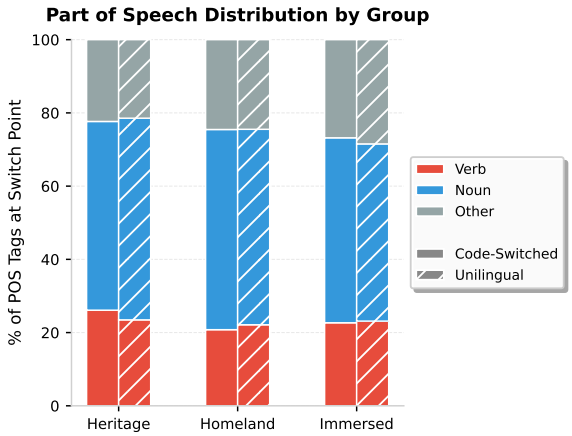


Figure 6: Distribution of POS tags at the switch point by bilingual group, drawn from the full matched dataset prior to model-specific filtering (e.g., exclusion of proper nouns and multi-word switches). POS distributions are broadly comparable across groups.

B.2 Sentence Matching Diagnostics

Table 4 reports diagnostics for the deterministic ranking procedure used to select a unilingual match when multiple candidates satisfied the filtering criteria. Lower ranks indicate higher-priority candidates in the selection process.

C Computational Methodology Details

C.1 Surprisal Formulations

For the masked model, each subword token w_k of the target word was masked in turn and the probability assigned to the original token at the masked position was extracted from the softmax distribution over the full vocabulary.

Stage	Sentence
Original (CS)	但係如果**international school** 好
Invalid Translation	但係如果國際學校**school** 好
Truncated	但係如果國際學校
POS Tags	CCONJ CCONJ NOUN

Table 3: The truncation pipeline: The original code-switched sentence contains at least two Cantonese words before the switch point. To satisfy the lexical constraint of no English words post-translation, the sentence is truncated after the last valid translation, in this case occurring at the switch point. This removes the untranslated English segments.

Table 4: Diagnostics for the deterministic sentence-matching procedure.

Matching Rank Statistic	Mean	Median	SD
Same-speaker match rank	0.893	0	2.569
Same-group match rank	5.790	3	8.401

Token log-probabilities were computed in log space for numerical stability. Any non-finite log-probability (i.e., underflow to $-\infty$) was treated as a precision-floor event, with the corresponding token surprisal capped at $-\log_2(\epsilon_{\text{float32}}) = 126.9$ bits rather than discarding the instance. Only instances with undefined values due to token–word alignment failures (NaNs) were excluded, along with their matched unilingual pairs.

C.2 Discourse Context Translation Quality

Translation can occasionally yield unusable outputs in multi-sentence context windows (e.g., empty or malformed generations, or outputs that still contain English tokens), which would reintroduce out-of-distribution tokens into the Cantonese context. To guard against this, we computed a word-level translation coverage ratio for each context window, defined as the proportion of words in the translated output that were neither untranslatable tokens (`UNKNOWN`, `UNK`) nor residual English words. Context windows consisting entirely of Cantonese sentences require no translation and receive a ratio of 1 by construction. Context windows with a coverage ratio < 0.3 were flagged as invalid; in practice, no context windows in the final dataset fell below this threshold.

D Generalized Mixed-Effects Logistic Regression Models

The following tables are the other four mixed-effects logistic regression models: (1) Table 5 is a model using surprisal from an autoregressive language model with no prior context, (2) Table 6 is a model using surprisal from the autoregressive model with three sentences of preceding context, (3) Table 7 is a model using surprisal from a masked language model with no prior context, and (4) Table 8 is a model using surprisal from a masked language model with three sentences of preceding context.

	Log-Odds Estimate	Std. Error	z score	p-value
Intercept	-2.05	0.23	-9.07	< 0.001
Word Length	0.83	0.11	7.89	< 0.001
POS: Noun	-0.22	0.08	-2.62	0.009
POS: Verb	0.10	0.10	1.08	0.279
Surprisal	0.03	0.01	4.20	< 0.001
Bilingual Group (Heritage)	0.10	0.24	0.41	0.683
Bilingual Group (Immersed)	-0.36	0.25	-1.47	0.143
Surprisal $\times$ Heritage	-0.01	0.01	-0.49	0.623
Surprisal $\times$ Immersed	0.02	0.01	1.48	0.138

Table 5: Mixed-effects logistic regression results from Model (1a) (autoregressive language model with no prior context) predicting CS probability. Higher surprisal was associated with a greater likelihood of CS.

	Log-Odds Estimate	Std. Error	z score	p-value
Intercept	-1.58	0.22	-7.14	< 0.001
Word Length	1.06	0.10	10.08	< 0.001
POS: Noun	-0.19	0.08	-2.28	0.022
POS: Verb	0.11	0.09	1.18	0.239
Surprisal	-0.02	0.01	-2.92	0.004
Bilingual Group (Heritage)	0.05	0.23	0.23	0.814
Bilingual Group (Immersed)	-0.11	0.23	-0.47	0.635
Surprisal $\times$ Heritage	-0.00	0.01	-0.43	0.665
Surprisal $\times$ Immersed	0.00	0.01	0.30	0.766

Table 6: Mixed-effects logistic regression results from Model (1b) (autoregressive language model with three sentences of preceding context) predicting CS probability. Higher surprisal was associated with a lower likelihood of CS.

	Log-Odds Estimate	Std. Error	z score	p-value
Intercept	-2.12	0.23	-9.26	< 0.001
Word Length	0.87	0.10	8.53	< 0.001
POS: Noun	-0.24	0.08	-2.89	0.004
POS: Verb	0.07	0.10	0.71	0.478
Surprisal	0.06	0.01	6.46	< 0.001
Bilingual Group (Heritage)	-0.07	0.21	-0.35	0.729
Bilingual Group (Immersed)	-0.24	0.21	-1.13	0.259
Surprisal $\times$ Heritage	0.01	0.01	0.45	0.651
Surprisal $\times$ Immersed	0.02	0.01	1.29	0.197

Table 7: Mixed-effects logistic regression results from Model (2a) (masked language model with no prior context) predicting CS probability. Higher surprisal was associated with a greater likelihood of CS.

	Log-Odds Estimate	Std. Error	z score	p-value
Intercept	-2.15	0.23	-9.26	< 0.001
Word Length	0.98	0.10	9.58	< 0.001
POS: Noun	-0.21	0.08	-2.47	0.013
POS: Verb	0.06	0.10	0.64	0.520
Surprisal	0.06	0.01	5.59	< 0.001
Bilingual Group (Heritage)	-0.19	0.20	-0.96	0.339
Bilingual Group (Immersed)	-0.02	0.19	-0.08	0.937
Surprisal $\times$ Heritage	0.03	0.02	1.75	0.080
Surprisal $\times$ Immersed	-0.01	0.01	-0.55	0.583

Table 8: Mixed-effects logistic regression results from Model (2b) (masked LM with three sentences of preceding context) predicting code-switching. Higher surprisal was associated with a greater likelihood of code-switching.

Non-literal Meaning Representation in the Brain during Naturalistic Listening

Zhengwu Ma, Yuhan Huang, Chengcheng Wang, Jixing Li*
Department of Linguistics and Translation, City University of Hong Kong,
{zhengwu.ma, yuhan.huang, chengcheng.wang}@my.cityu.edu.hk,
jixingli@cityu.edu.hk

Abstract

Naturalistic language comprehension often involves interpretations that go beyond literal meaning. In continuous narratives, literal and non-literal meanings are tightly intertwined, making them difficult to distinguish computationally. Here, we combined literal sentence representations and human-annotated non-literal interpretations for model-brain alignment. Using fMRI data recorded during passive listening to the Chinese version of *The Little Prince*, we annotated sentences containing non-literal meaning with human-written interpretations of their implied meaning. We then derived the literal and non-literal representations from LLaMA3.1-8B and evaluated their correspondence with neural activity using whole-brain encoding models. Literal representations aligned strongly with left-lateralized frontotemporal regions, whereas non-literal interpretations showed broader right-hemisphere involvement. Combining the two further improved encoding performance in the bilateral temporal and dorsal frontal cortices, suggesting that naturalistic comprehension engages complementary levels of meaning.

1 Introduction

When we listen to a story, we do not simply process the words that are explicitly stated. We constantly infer "implied" messages based on context, such as metaphors, symbolic meanings, or the speaker's intentions. This suggests at least two co-existing meaning dimensions in everyday understanding: literal meaning, grounded in the surface sentence, and non-literal (inferred) meaning, reflecting what the sentence is taken to convey in context (Grice, 1975; Zwaan and Radvansky, 1998). A central question is how literal and non-literal meaning relate to patterns of brain activity during naturalistic comprehension.

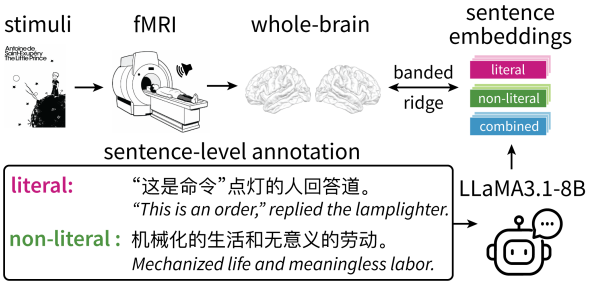


Figure 1: Aligning literal and non-literal meaning embeddings with whole-brain fMRI data during naturalistic listening in the Chinese version of *The Little Prince*.

Much of what we know about non-literal interpretation in the brain comes from neuroimaging studies that use carefully controlled contrasts, often focusing on specific phenomena such as metaphor or irony (Rapp et al., 2004; Eviatar and Just, 2006; Zempleni et al., 2007). These studies have been invaluable for isolating particular processes, but their simplified stimuli and task settings do not fully capture how meaning unfolds in the naturalistic context.

In real narratives, multiple cues co-occur, and information accumulates over time. Neural responses therefore reflect moment-to-moment processing and integration in an extended context (Hasson et al., 2008; Lerner et al., 2011). This makes it unclear how findings from controlled paradigms carry over to naturalistic story comprehension. What we still lack is a way to compare literal text and inferred interpretations for the same narrative content in a matched manner.

To make such a matched comparison possible, we need predictors that are quantifiable, time-aligned, and directly comparable across meaning dimensions. Recent large language models (LLMs) are well suited to this role: they cover a wide range of language tasks, and their representations have been shown to systematically relate to neural responses during language comprehension (Schrimpf

*Corresponding author.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 244–257
July 3-4, 2026 ©2026 Association for Computational Linguistics

et al., 2021; Goldstein et al., 2022; Caucheteux and King, 2022; Caucheteux et al., 2023; Gao et al., 2024, 2025; Kumar et al., 2024).

Here, we introduce a measurement framework for disentangling multiple meaning dimensions during naturalistic language comprehension. Using whole-brain functional magnetic resonance imaging (fMRI) recorded during continuous listening to the Chinese audiobook of *The Little Prince*, we aligned the narrative with the neural time series at the sentence level and constructed paired predictors for the same content (see Figure 1). For narratives, we used two matched texts: the original sentence and a human-annotated non-literal interpretation that summarized what the sentence is understood to convey in the story. Using LLMs, we extracted sentence-level features for both literal and non-literal sentences, and then evaluated these predictors in whole-brain encoding models. Our results reveal:

- Literal and non-literal meaning were related to both shared and dissociable cortical response patterns during continuous story comprehension, with a tendency for non-literal signals to extend more into right-hemisphere regions.
- We introduce a feasible measurement for separating language comprehension in naturalistic narratives, enabling comparisons in brain encoding analysis.
- We provide sentence-level non-literal annotations aligned to the original narrative text, enabling matched comparisons of surface content and inferred meaning in future work.

2 Related Work

2.1 Neural basis of literal and non-literal language comprehension

Language comprehension is supported by a left-lateralized fronto-temporal network, with additional bilateral involvement in speech and semantic processing (Hagoort, 2013; Hickok and Poeppel, 2007). Within this network, the left temporal and parietal areas, including the middle temporal gyrus (MTG), the superior temporal gyrus (STG) and the angular gyrus (AG), handle accessing and representing word meanings (Binder et al., 2009), whereas the inferior frontal gyrus (IFG) contributes to integration and control during sentence processing (Hagoort, 2013; Matchin and Hickok, 2020). This network is often framed as interactions between a transmodal semantic hub in the anterior temporal lobes (ATL) and fronto-temporal mechanisms that build hierarchical meaning from simpler constituents (Patterson et al., 2007; Ralph et al., 2017; Pallier et al., 2011; Lau et al., 2008). This framework serves as a general model of meaning construction, but it is less specific about how to dissociate complex context-driven meaning in naturalistic narratives.

Work on context-dependent interpretation suggests that going beyond the surface sentence can recruit broader integrative and control processes. The right hemisphere, for example, has been argued to maintain broader contextual information or diffuse semantic activation that supports remote associations (Beeman et al., 1994; Jung-Beeman, 2005). Studies of metaphors and symbolic expressions often report additional recruitment of regions in the right hemisphere, such as rIFG and rMTG, and when interpretive demands increase, while dorsal frontal areas (e.g., middle/superior frontal areas) are activated for monitoring and cognitive control (Bohrn et al., 2012; Rapp et al., 2012). Interpretation related to social intent or affect can also involve salience-related regions such as the anterior insula and the anterior cingulate cortex (Citron and Goldberg, 2014; Seeley et al., 2007). However, much of this evidence comes from controlled contrasts that isolate specific phenomena (e.g., Rapp et al., 2004; Eviatar and Just, 2006; Zempleni et al., 2007), leaving open how literal content and non-literal interpretations co-exist, and whether they can be disentangled during continuous naturalistic narrative comprehension.

2.2 Model-derived language features for naturalistic encoding

Naturalistic encoding work has shown that semantic features derived from a text-based representational space can be linked to distributed cortical responses during continuous story listening (Huth et al., 2016). This framework is widely used to map semantic organization onto the brain cortex under naturalistic stimulation, and for our question it is crucial to use sentence-level features that capture meaning in context. Modern transformer-based language models provide a practical shared representational space for this purpose because they generate contextualized representations that can be extracted at the sentence-level or even the discourse-level in a uniform way (Vaswani et al., 2017; Devlin et al.,

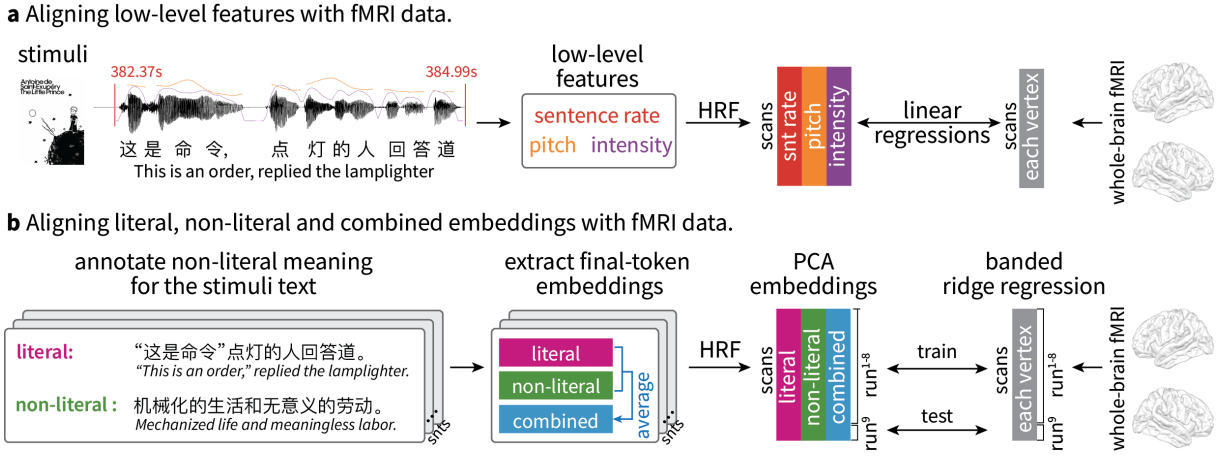


Figure 2: Methods overview. **a** Aligning the low-level regressors, sentence rate, pitch, and intensity with whole-brain fMRI data using GLMs. **b** Aligning literal, non-literal and combined sentence representations with whole-brain fMRI data using banded ridge regression.

2019; Radford et al., 2018). Recent model-derived features of LLMs have shown robust correspondence with neural responses during naturalistic language processing (Caucheteux et al., 2023; Kumar et al., 2024; Goldstein et al., 2024).

LLM-derived representations are sensitive to the amount of input context. Recent work has explicitly manipulated the amount of preceding context and shown that alignment can change substantially with the window size of the context (Yu et al., 2024). Long-context inputs have also been used to better capture complex information integration (Goldstein et al., 2025a,b). Here, in this work, we intend to apply a matched sentence-level comparison across meaning dimensions. We extracted sentence-locked representations by feeding each sentence on its own, without adding extra discourse context, to keep the comparison clean and interpretable. We then contrasted features derived from the narrative's surface sentence with features derived from explicit interpretation annotations. This setup motivated our paired-predictor design, enabling comparable tests of literal versus non-literal meaning during naturalistic comprehension.

3 Methods

3.1 Non-literal meaning annotations

We adopted the classic storybook *The Little Prince* as our experimental text because its language is concise, while its intended meanings are often rich and implicit. The Chinese version of *The Little Prince* audiobook contains 15,603 words distributed across 1,289 sentences (mean sentence length = 12.10 words, SD = 9.37). We operationalized *non-literal meaning* as sentences whose intended interpretation involves metaphorical, symbolic, inferential, or otherwise implied meanings that extend beyond their surface lexical-semantic composition.

Due to the lack of an established dataset with non-literal meaning annotations, we recruited two native Mandarin speakers with formal training in linguistics to annotate all sentences for the presence or absence of non-literal meaning. For sentences identified as non-literal, annotators additionally provided a written paraphrase of the intended non-literal interpretation. Inter-annotator disagreements were resolved through discussion until consensus was reached, ensuring annotation reliability. Detailed annotation instructions are provided in Section 6.

A total of 429 sentences (33.3% of the total sentences; mean sentence length = 13.67 words, SD = 10.25) were labeled as containing non-literal meaning. For the following analyses, we only focused on these 429 sentence pairs which both contain literal and non-literal meanings. Representative examples of annotated literal and non-literal contrasts are shown in Table 1.

3.2 fMRI data

We analyzed the Chinese subset of a publicly available naturalistic listening fMRI dataset (Li et al., 2022), including 35 native Mandarin speakers (15 females; age = 19.3 ± 1.6 years). During scanning, participants passively listened to the complete Chinese audiobook of *The Little Prince* (approximately 99 minutes) in a single session divided into

ID	Literal Meaning	Non-literal Meaning
1	“这是命令,” 点灯的人回答道。 “This is an order,” replied the lamplighter.	机械化的生活和无意义的劳动。 Mechanized life and meaningless labor.
2	他的那朵花曾对他说，她是整个宇宙中独一无二的一种花。 His flower had once told him that she was a flower unique in all the universe.	小王子怀念爱情，感到心疼后悔。 The little prince misses his love, feeling heartache and regret.
3	当我们默默地走了好几个小时以后，天黑了下来，星星开始发出光亮。 When we had trudged along for several hours, in silence, the darkness fell, and the stars began to come out.	星星的亮起提醒人们即使在最艰难的情况下也总有希望存在。 The lighting up of the stars reminds people that even in the most difficult situations, there is always hope.

Table 1: Examples of literal and non-literal meaning annotations from the Chinese version of *The Little Prince*.

nine runs of approximately 10 minutes each. The preprocessed volumetric data were reconstructed onto a “fsaverage4” surface-based template. The fMRI signals are z-scored across the time dimension for each participant, surface vertex and session independently. Details for fMRI data acquisition and preprocessing procedure are provided in Appendix A.

3.3 Model

To quantify sentence-level representational features of literal and non-literal meanings, we extracted final-token embeddings from the open-source large language model LLaMA3.1 (Grattafiori et al., 2024) for both the original sentences and their annotated interpretations. LLaMA3.1 has been widely used in model-brain alignment research (e.g., Binz et al., 2025; Gao et al., 2025; An et al., 2025) and offers strong multilingual capabilities. To manage computational resources (see Appendix B), we used the 8B-size model.

3.4 Representation differences between literal and non-literal meanings

Linear probing from literal embeddings to non-literal embeddings. We first tested whether non-literal embeddings can be recovered from literal embeddings via a simple linear mapping. For each model layer, we trained a ridge regression model to predict the non-literal embedding vector from the corresponding literal embedding vector. Prediction performance was evaluated with 5-fold cross-validated coefficient of determination (R^2). As a permutation control, we randomly shuffled the pairing between literal and non-literal embeddings and repeated the same layer-wise probing procedure.

Dimensionality reduction of literal and non-literal representations. We applied principal component analysis (PCA) to the sentence-level embeddings for the literal and non-literal conditions. For each sentence, we first averaged the model hidden states across layers to obtain a single 4096-dimensional vector, and then projected these vectors onto the first three principal components. We visualized the resulting 3D projections to examine the global geometry and separability of literal vs. non-literal representations.

3.5 Aligning low-level features with whole-brain fMRI data

We included two acoustic features and one sentence-level regressor: **pitch**, **intensity**, and **sentence rate**, which are known to correlate with activity in core language-related regions and served as baseline predictors against which the contribution of sentence-level semantic representation was evaluated (Momenian et al., 2024; Wang et al., 2025; Li et al., 2022). Pitch and root-mean-square (RMS) intensity were computed from the Chinese audiobook at 10 ms resolution using the Voicebox toolbox[1]. Sentence rate was modeled as a binary regressor marking the offset of each sentence in the audiobook.

For each run, these low-level regressors were convolved with the canonical hemodynamic response function (HRF). The run-wise convolved regressors were then concatenated across runs, z-scored, and fit to each subject’s concatenated nine-run fMRI time series at each cortical vertex using ordinary least squares (OLS), as illustrated in Figure 2a.

[1] http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

3.6 Aligning literal and non-literal embeddings with whole-brain fMRI data

Identify the best-performing layer. We performed layer-wise encoding analyses at each cortical vertex using the extracted literal and non-literal embeddings. For each layer and condition (literal, non-literal), we constructed a time series of sentence-level embeddings, time-locked to each annotated sentence's offset. This embedding time series was then convolved with the HRF to match the temporal resolution of the fMRI data and used as predictors in vertex-wise ridge regression models. Models were trained on embeddings and fMRI responses concatenated across the first eight runs and evaluated on the held-out ninth run. Encoding performance was quantified as the Pearson correlation (r) between the predicted and observed fMRI time series in the test run. To identify the best-performing layer, we averaged r across vertices to obtain a single whole-brain encoding score for each layer.

Banded ridge regression. We next used banded ridge (multi-kernel) regression (Dupré la Tour et al., 2022) to jointly model fMRI responses using literal and non-literal representations. We selected embeddings from the best-performing LLaMA3.1 layers identified in the layer-wise encoding analysis and fit multi-kernel ridge models with Himalaya's MULTIPLEKERNELRIDGECV. Based on these selections, we constructed three predictor sets: (i) literal sentence embeddings, (ii) non-literal sentence embeddings, and (iii) a *combined* embedding defined as the average of the best-layer literal and non-literal embeddings. Each predictor set was reduced from 4096 to 100 dimensions using PCA, and run-wise predictors were concatenated across the first eight runs to form three separate linear kernels.

The regression models were trained in the first eight concatenated runs and evaluated in the ninth run. For each vertex and participant, kernel-specific regularization weights were selected via grid search with nested cross-validation over 10 values log-spaced from 10^0 to 10^{20} within the training data (Gao et al., 2025; Goldstein et al., 2025b; Huth et al., 2016). Kernel-specific performance was obtained by splitting each kernel's prediction and computing the Pearson correlation between that kernel's predicted time series and the observed vertex-wise fMRI time series in the test run. (see Figure 2b and Appendix C for regression details).

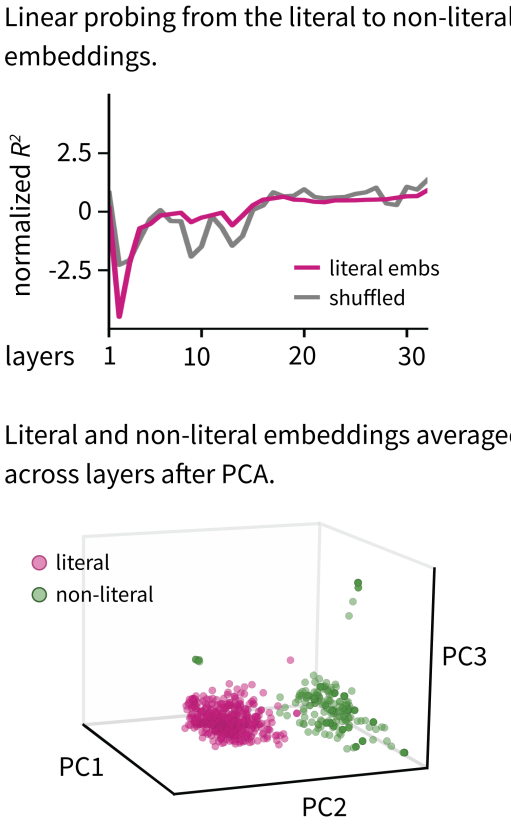


Figure 3: a Layer-wise linear probing from literal to non-literal embeddings using ridge regression. **b** PCA visualization of sentence-level embeddings after averaging across layers. Each point is a sentence projected onto the first three principal components.

3.7 Statistical significance testing

Linear probe statistical testing. We first averaged the 5-fold R^2 scores for the literal→non-literal mapping and for the shuffled-pairing control. We then compared probing performance between the two conditions using a two-tailed paired-sample t-test.

GLM and banded ridge regressions. At the group level, vertex-wise GLM regression coefficients (β) and banded ridge encoding performance (Pearson's r) were first z-scored and evaluated with one-sample, one-tailed cluster-based permutation tests (Maris and Oostenveld, 2007) with 10,000 permutations. Clusters were formed from statistics corresponding to a p-value less than 0.05, and only clusters spanning a minimum of 20 vertices were included in the analysis.

Contrasts between literal and non-literal encoding maps. To compare the encoding performance between conditions, we computed vertex-wise contrast maps by subtracting the non-literal embedding $r$map from the corresponding literal

embedding *r*map (literal > non-literal). The reverse contrast (non-literal > literal) was computed conversely. The resulting contrast maps were z-scored and evaluated using the same cluster-based permutation procedure described above. All statistical analyses were performed using custom Python codes, making heavy use of the Scipy (v1.12.0; Virtanen et al., 2020), MNE (v1.10.2; Gramfort et al., 2014) and Eelbrain (v0.41; Brodbeck et al., 2023) packages.

4 Results

4.1 Representational differences between literal and non-literal embeddings

Linear probing from literal to non-literal embeddings. We found no significant difference in probing performance between the true literal→non-literal mapping and the shuffled control. Across layers, ridge-based linear probes from literal to non-literal embeddings did not outperform the shuffled baseline (Figure 3a), indicating that non-literal embeddings are not reliably predictable from literal embeddings using a simple linear mapping in this embedding space.

Dimensionality reduction of literal and non-literal embeddings. PCA visualization of two embeddings showed that literal and non-literal representations form separable clusters in the low-dimensional space (Figure 3b), suggesting systematic differences in their global geometry. Together with the null linear-probing result, this pattern indicates that the two representations are separable in embedding space, while their relationship is not well captured by a single linear mapping.

4.2 Encoding performance across LLaMA3.1 model layers

The best-performing encoding layers for literal and non-literal embeddings were layer 11 and layer 21, respectively (see Figure 4a). This pattern is consistent with prior model-brain alignment work showing that mid-to-late layers often yield the strongest encoding performance.

Correlations among the literal and non-literal sentence embeddings from their best-performing layers, as well as the combined embedding defined as their average, are shown in Figure 4b. Literal and non-literal embeddings exhibited low similarity (r=0.081), consistent with the representational differences above. The combined embedding was strongly correlated with the non-literal embedding

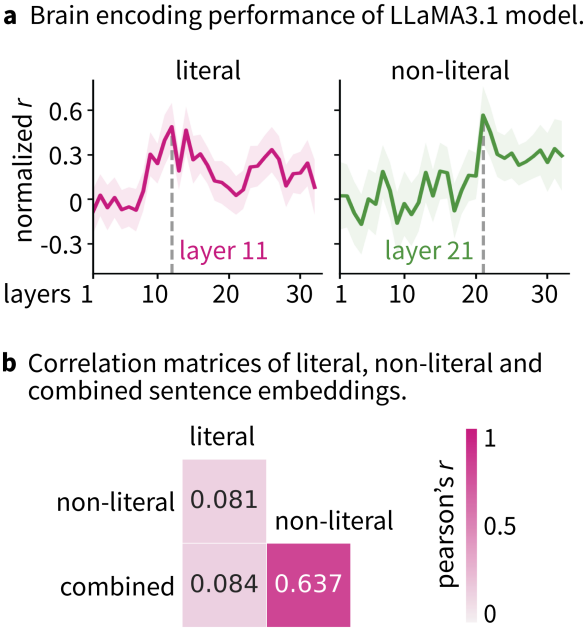


Figure 4: **a** Layer-wise encoding performance of LLaMA3.1 against fMRI data using literal and non-literal embeddings. **b** Correlation matrices of literal, non-literal, and combined embeddings from their best-performing LLaMA3.1 layers.

(r=0.637) but only weakly correlated with the literal embedding (r=0.084).

4.3 Brain regions associated with low-level features

Correlations among the HRF-convolved sentence rate, pitch, and intensity regressors are shown in Figure 5a. The strongest association was between pitch and intensity (r=0.433), whereas correlations involving two acoustic features and sentence rate were close to zero (both r<0.02).

We observed significant bilateral effects of the low-level features in temporal and frontal regions, consistent with prior work on speech-related processing (Li et al., 2022; Momenian et al., 2024; Wang et al., 2025). The acoustic regressors (pitch and intensity) showed robust effects in the temporal cortex: pitch exhibited a significant association in the bilateral ATL and the MTG, with a stronger effect in the right hemisphere, while intensity was primarily associated with the bilateral STG. Sentence rate was associated with responses in left ATL, left IFG, bilateral STG, MTG, and ventromedial prefrontal cortex (vmPFC). Detailed clusters and corresponding statistics for the low-level features are reported in Figure 5b and Table 2.

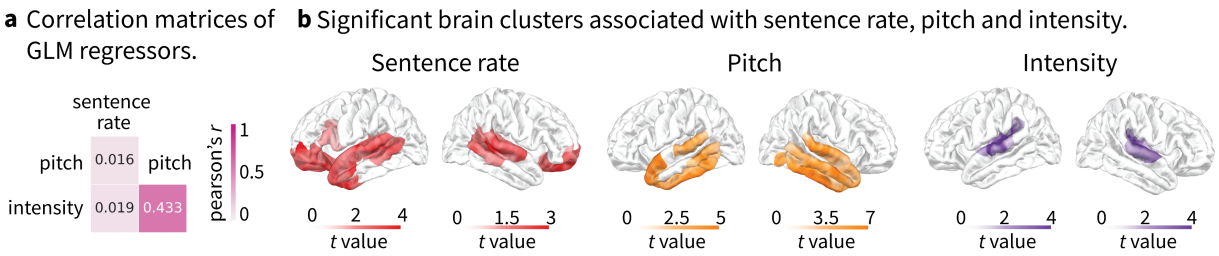


Figure 5: **a** Correlation matrices among three low-level features. **b** Significant brain clusters associated with sentence rate, pitch and intensity.

Embedding group	Left hemisphere			Right hemisphere		
	N vertices	p	Cohen's d	N vertices	p	Cohen's d
Sentence rate	758	0.0061	3.7642	722	0.0199	3.1042
Pitch	257	0	2.4034	452	0	2.3383
Intensity	249	0.0189	2.4325	222	0.0374	3.1270

Table 2: Cluster statistics for sentence rate, pitch, and intensity effects from the GLM analysis.

4.4 Brain regions associated with literal, non-literal and combined embeddings

We observed widespread frontal and temporal clusters across the literal, non-literal, and combined conditions, with lateralization patterns that differed by condition (Figure 6a).

In the literal condition, left-lateralized effects spanned canonical language regions from frontal to temporoparietal areas, including the IFG, insula, STG/MTG, and AG. Smaller right-hemisphere clusters were also detected in the frontal/insular cortices and temporal regions, including IFG, insula, STG/MTG.

The non-literal condition showed a more right-lateralized pattern. Only a small cluster was observed in the left insula, whereas the right hemisphere exhibited widespread effects across frontal and temporal regions, including the precentral gyrus (PrG), caudal middle frontal gyrus (cMFG), IFG, and STG.

In the combined embedding, the effects were more prominent in the right hemisphere overall. Left-hemisphere effects were observed mainly in superior frontal gyrus (SFG), STG/MTG. In the right hemisphere, the effects extended across frontal, insular, and temporal cortices, including SFG, insula, STG/MTG, and the temporal pole. Table 3 reports detailed literal, non-literal and combined cluster statistics.

4.5 Brain regions associated with contrasts between literal and non-literal embeddings

Contrast analyses also revealed lateralized differences between conditions: the literal > non-literal contrast was strongest in the left frontal and temporal regions, whereas the non-literal > literal contrast was strongest in the right frontal regions (see Figure 6b). Specifically, the literal > non-literal contrast yielded significant clusters in the left IFG and STG, while the non-literal > literal contrast yielded marginally significant clusters confined to the right frontal cortex, with peaks in the precentral gyrus, SFG, and cMFG. Detailed statistics are reported in Table 3.

5 Discussion

A large body of model–brain alignment work shows that contextual embeddings derived from large language models predict neural activity during naturalistic language comprehension (e.g., Gao et al., 2024, 2025; Goldstein et al., 2022; Huth et al., 2016; Toneva et al., 2022; Schrimpf et al., 2021). Most prior studies, however, use embeddings derived directly from surface text and therefore primarily index literal meaning. Here, we offered a measurement framework by aligning fMRI responses to embeddings derived from manually literal and annotated non-literal interpretations of the same narrative content. We found literal and non-literal representations derived from LLM were clearly distinct in embedding space, and non-literal representations could not be reliably recovered from literal embeddings with a simple linear probe.

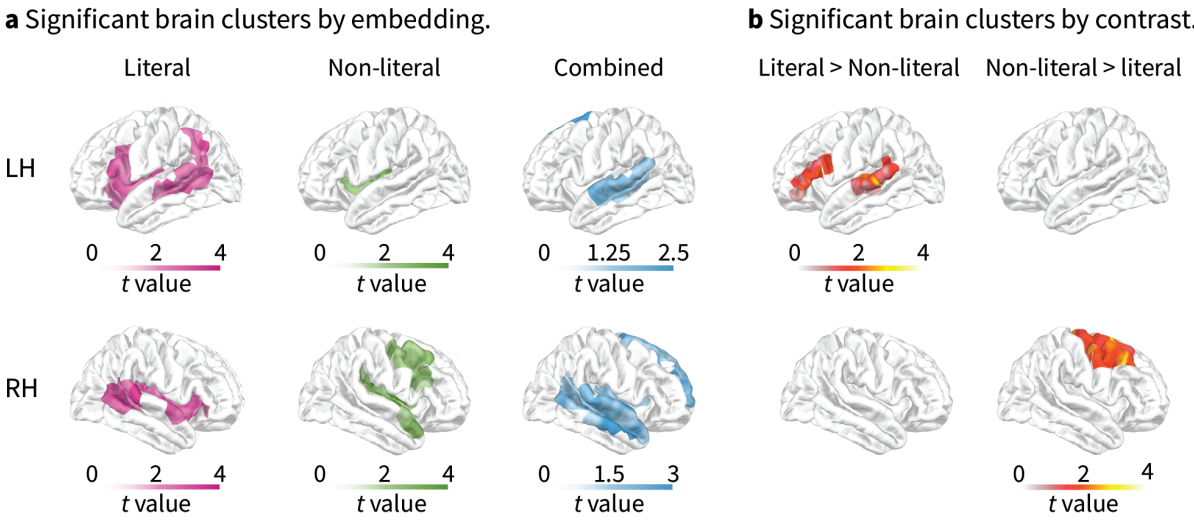


Figure 6: **a** Significant brain clusters associated with literal, non-literal and combined embeddings. **b** Significant brain clusters for literal > non-literal and non-literal > literal contrasts.

Embedding group	Left hemisphere			Right hemisphere		
	N vertices	p	Cohen's d	N vertices	p	Cohen's d
Literal	123	0.001	6.2609	112	0.0007	6.5042
Non-literal	30	0.0091	5.2102	129	0.0056	3.8235
Combined	55	0.0145	3.8592	212	0.0052	4.3092
Contrast						
Literal > Non-literal	43	0.002	2.5873	/	/	/
Non-literal > Literal	/	/	/	76	0.0631	3.8239

Table 3: Cluster statistics for literal, non-literal, and combined sentence embeddings from banded ridge regression.

In the brain, both literal and non-literal embeddings significantly predicted widespread responses, but with condition-dependent spatial patterns. Literal embeddings aligned most strongly with left-lateralized temporal and frontal regions typically implicated in core semantic processing, consistent with prior model-brain alignment findings. Whereas non-literal embeddings showed broader right-hemisphere involvement in frontal and temporal cortices, consistent with accounts linking the right hemisphere to integrative and interpretive aspects of meaning (Jung-Beeman, 2005; Binder et al., 2009; Rapp et al., 2004; Fedorenko et al., 2024). Importantly, combining the best-layer literal and non-literal embeddings improved encoding performance and produced bilateral effects, suggesting that neural responses during narrative comprehension reflect multiple levels of meaning in parallel rather than a single surface-based semantic representation.

It is also worth noting that standard LLM training objectives are not designed to explicitly separate surface-form semantics from interpretive or non-literal dimensions of meaning. Solely relying on literal-derived embeddings may miss aspects of meaning that are critical for naturalistic comprehension.

6 Conclusion

We investigated whether model-derived representations of non-literal meaning align with neural activity during naturalistic narrative comprehension. Embeddings derived from manually annotated non-literal interpretations significantly predicted brain responses, and combining literal and non-literal representations improved encoding performance in bilateral temporal and frontal regions, suggesting that comprehension is sensitive to both compositional and interpretive information.

However, LLM-derived embeddings from the surface form alone did not fully reflect non-literal meaning: literal embeddings did not reliably predict non-literal embeddings with a simple linear mapping, and the two representations were separable in embedding space. These findings motivate incorporating explicit interpretive semantic dimensions in model-brain alignment to better capture naturalistic language understanding.

Limitations

Several limitations should be acknowledged. First, our non-literal embeddings were derived from manually annotated interpretations. While this ensures interpretive specificity, non-literal meaning is inherently context-dependent and may vary across individuals. Future work could develop computational approaches to model interpretive meaning at scale and assess inter-annotator and inter-reader variability more directly.

Second, our definition of *non-literal meaning* pooled multiple types of implied meaning (e.g., metaphorical, symbolic, and inferential readings) that may recruit partially distinct neural mechanisms. Current analyses do not dissociate these subtypes, and future studies should test whether categories such as metaphor, irony, and implicature show distinct alignment patterns.

Finally, we analyzed only the Chinese subset of a multilingual naturalistic fMRI dataset (Li et al., 2022). This controlled the annotation pipeline, but limited claims about cross-linguistic generalizability. Given that non-literal meaning is shaped by language structure and cultural conventions, future work should leverage the English and French data to examine shared versus language-specific neural signatures of interpretive meaning.

Ethics Statement

The authors declare no competing interests. The non-literal meanings were manually annotated by two trained annotators following the instructions to "identify and specify the non-literal interpretation of each sentence whenever applicable." Annotators were recruited through advertisements and were compensated for their work in accordance with local rates; they reported no conflicts of interest. The annotation task involved no risks.

The fMRI dataset used in the analysis is publicly available and does not contain sensitive content, such as personal information. The adaptation and use of the fMRI dataset were conducted in accordance with its license.

The model states of LLaMA3.1-8B are utilized solely for research purposes, aligning with its intended use.

References

Jingmin An, Yilong Song, Ruolin Yang, Nai Ding, Lingxi Lu, Yuxuan Wang, Wei Wang, Chu Zhuang, Qian Wang, and Fang Fang. 2025. Hierarchical frequency tagging probe (HFTP): a unified approach to investigate syntactic structure representations in large language models and the human brain. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Mark Beeman, Rhonda B. Friedman, Jordan Grafman, Enrique Perez, Sherri Diamond, and Miriam Beadle Lindsay. 1994. Summation Priming and Coarse Semantic Coding in the Right Hemisphere. *Journal of Cognitive Neuroscience*, 6(1):26–45.

Jeffrey R. Binder, Rutvik H. Desai, William W. Graves, and Lisa L. Conant. 2009. Where is the semantic system? a critical review and meta-analysis of 120 functional neuroimaging studies. *Cerebral Cortex*, 19(12):2767–2796.

Marcel Binz, Elif Akata, Matthias Bethge, Franziska Brändle, Fred Callaway, Julian Coda-Forno, Peter Dayan, Can Demircan, Maria K. Eckstein, Noémi Éltető, Thomas L. Griffiths, Susanne Haridi, Akshay K. Jagadish, Li Ji-An, Alexander Kipnis, Sreejan Kumar, Tobias Ludwig, Marvin Mathony, Marcelo Mattar, Alireza Modirshanechi, Surabhi S. Nath, Joshua C. Peterson, Milena Rmus, Evan M. Russek, Tankred Saanum, Johannes A. Schubert, Luca M. Schulze Buschoff, Nishad Singhi, Xin Sui, Mirko Thalmann, Fabian J. Theis, Vuong Truong, Vishaal Udandarao, Konstantinos Voudouris, Robert Wilson, Kristin Witte, Shuchen Wu, Dirk U. Wulff, Huadong Xiong, and Eric Schulz. 2025. A foundation model to predict and capture human cognition. *Nature*, 644(8078):1002–1009.

Isabel C. Bohrn, Ulrike Altmann, and Arthur M. Jacobs. 2012. Looking at the brains behind figurative language—A quantitative meta-analysis of neuroimaging studies on metaphor, idiom, and irony processing. *Neuropsychologia*, 50(11):2669–2683.

Christian Brodbeck, Proloy Das, Marlies Gillis, Joshua P Kulasingham, Shohini Bhattasali, Phoebe Gaston, Philip Resnik, and Jonathan Z Simon. 2023. Eelbrain, a Python toolkit for time-continuous analysis with temporal response functions. *eLife*, 12:e85012.

Charlotte Caucheteux, Alexandre Gramfort, and Jean-Rémi King. 2023. Evidence of a predictive coding hierarchy in the human brain listening to speech. *Nature Human Behaviour*, 7(3):430–441.

Charlotte Caucheteux and Jean-Rémi King. 2022. Brains and algorithms partially converge in natural language processing. *Communications Biology*, 5(1):134.

Francesca M. M. Citron and Adele E. Goldberg. 2014. Metaphorical Sentences Are More Emotionally Engaging than Their Literal Counterparts. *Journal of Cognitive Neuroscience*, 26(11):2585–2595.

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. Association for Computational Linguistics.

Tom Dupré la Tour, Michael Eickenberg, Anwar O. Nunez-Elizalde, and Jack L. Gallant. 2022. Feature-space selection with banded ridge regression. *NeuroImage*, 264:119728.

Oscar Esteban, Rastko Ciric, Karolina Finc, Ross W. Blair, Christopher J. Markiewicz, Craig A. Moodie, James D. Kent, Mathias Goncalves, Elizabeth DuPre, Daniel E. P. Gomez, Zhifang Ye, Taylor Salo, Romain Valabregue, Inge K. Amlien, Franziskus Liem, Nir Jacoby, Hrvoje Stojić, Matthew Cieslak, Sebastian Urchs, Yaroslav O. Halchenko, Satrajit S. Ghosh, Alejandro De La Vega, Tal Yarkoni, Jessey Wright, William H. Thompson, Russell A. Poldrack, and Krzysztof J. Gorgolewski. 2020. Analysis of task-based functional MRI data preprocessed with fMRIPrep. *Nature Protocols*, 15(7):2186–2202.

Zohar Eviatar and Marcel Adam Just. 2006. Brain correlates of discourse processing: An fMRI investigation of irony and conventional metaphor comprehension. *Neuropsychologia*, 44(12):2348–2359.

Evelina Fedorenko, Anna A. Ivanova, and Tamar I. Regev. 2024. The language network as a natural kind within the broader landscape of the human brain. *Nature Reviews Neuroscience*, 25(5):289–312.

Changjiang Gao, Jixing Li, Jiajun Chen, and Shujian Huang. 2024. Measuring meaning composition in the human brain with composition scores from large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11295–11308. Association for Computational Linguistics.

Changjiang Gao, Zhengwu Ma, Jiajun Chen, Ping Li, Shujian Huang, and Jixing Li. 2025. Increasing alignment of large language models with language processing in the human brain. *Nature Computational Science*, 5(11):1080–1090.

Ariel Goldstein, Avigail Grinstein-Dabush, Mariano Schain, Haocheng Wang, Zhuoqiao Hong, Bobbi Aubrey, Samuel A. Nastase, Zaid Zada, Eric Ham, Amir Feder, Harshvardhan Gazula, Eliav Buchnik, Werner Doyle, Sasha Devore, Patricia Dugan, Roi Reichart, Daniel Friedman, Michael Brenner, Avinatan Hassidim, Orrin Devinsky, Adeen Flinker, and Uri Hasson. 2024. Alignment of brain embeddings and artificial contextual embeddings in natural language points to common geometric patterns. *Nature Communications*, 15(1):2768.

Ariel Goldstein, Eric Ham, Mariano Schain, Samuel A. Nastase, Bobbi Aubrey, Zaid Zada, Avigail Grinstein-Dabush, Harshvardhan Gazula, Amir Feder, Werner Doyle, Sasha Devore, Patricia Dugan, Daniel Friedman, Michael Brenner, Avinatan Hassidim, Yossi Matias, Orrin Devinsky, Noam Siegelman, Adeen Flinker, Omer Levy, Roi Reichart, and Uri Hasson. 2025a. Temporal structure of natural language processing in the human brain corresponds to layered hierarchy of large language models. *Nature Communications*, 16(1):10529.

Ariel Goldstein, Haocheng Wang, Leonard Niekerken, Mariano Schain, Zaid Zada, Bobbi Aubrey, Tom Sheffer, Samuel A. Nastase, Harshvardhan Gazula, Aditi Singh, Aditi Rao, Gina Choe, Catherine Kim, Werner Doyle, Daniel Friedman, Sasha Devore, Patricia Dugan, Avinatan Hassidim, Michael Brenner, Yossi Matias, Orrin Devinsky, Adeen Flinker, and Uri Hasson. 2025b. A unified acoustic-to-speech-to-language embedding space captures the neural basis of natural language processing in everyday conversations. *Nature Human Behaviour*, 9(5):1041–1055.

Ariel Goldstein, Zaid Zada, Eliav Buchnik, Mariano Schain, Amy Price, Bobbi Aubrey, Samuel A. Nastase, Amir Feder, Dotan Emanuel, Alon Cohen, Aren Jansen, Harshvardhan Gazula, Gina Choe, Aditi Rao, Catherine Kim, Colton Casto, Lora Fanda, Werner Doyle, Daniel Friedman, Patricia Dugan, Lucia Melloni, Roi Reichart, Sasha Devore, Adeen Flinker, Liat Hasenfratz, Omer Levy, Avinatan Hassidim, Michael Brenner, Yossi Matias, Kenneth A. Norman, Orrin Devinsky, and Uri Hasson. 2022. Shared computational principles for language processing in humans and deep language models. *Nature Neuroscience*, 25(3):369–380.

Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Lauri Parkkonen, and Matti S. Hämäläinen. 2014. MNE software for processing MEG and EEG data. *NeuroImage*, 86:446–460.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan

Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin,

Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The Llama 3 herd of models. *Preprint*, arxiv:2407.21783 [cs].

H. P. Grice. 1975. Logic and Conversation. In *Speech Acts*, pages 41–58. Brill.

Peter Hagoort. 2013. MUC (Memory, Unification, Control) and beyond. *Frontiers in Psychology*, 4.

Uri Hasson, Eunice Yang, Ignacio Vallines, David J. Heeger, and Nava Rubin. 2008. A Hierarchy of Temporal Receptive Windows in Human Cortex. *Journal of Neuroscience*, 28(10):2539–2550.

Gregory Hickok and David Poeppel. 2007. The cortical organization of speech processing. *Nature Reviews Neuroscience*, 8(5):393–402.

Alexander G. Huth, Wendy A. de Heer, Thomas L. Griffiths, Frédéric E. Theunissen, and Jack L. Gallant. 2016. Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature*, 532(7600):453–458.

Mark Jung-Beeman. 2005. Bilateral brain processes for comprehending natural language. *Trends in Cognitive Sciences*, 9(11):512–518.

Sreejan Kumar, Theodore R. Sumers, Takateru Yamakoshi, Ariel Goldstein, Uri Hasson, Kenneth A. Norman, Thomas L. Griffiths, Robert D. Hawkins, and Samuel A. Nastase. 2024. Shared functional specialization in transformer-based language models and the human brain. *Nature Communications*, 15(1):5523.

Ellen F. Lau, Colin Phillips, and David Poeppel. 2008. A cortical network for semantics: (de)constructing the N400. *Nature Reviews Neuroscience*, 9(12):920–933.

Yulia Lerner, Christopher J. Honey, Lauren J. Silbert, and Uri Hasson. 2011. Topographic Mapping of a Hierarchy of Temporal Receptive Windows Using a Narrated Story. *Journal of Neuroscience*, 31(8):2906–2915.

Jixing Li, Shohini Bhattasali, Shulin Zhang, Berta Franzluebbers, Wen-Ming Luh, R. Nathan Spreng, Jonathan R. Brennan, Yiming Yang, Christophe Pallier, and John Hale. 2022. Le petit prince multilingual naturalistic fmri corpus. *Scientific Data*, 9(1):530.

Eric Maris and Robert Oostenveld. 2007. Nonparametric statistical testing of EEG- and MEG-data. *Journal of Neuroscience Methods*, 164(1):177–190.

William Matchin and Gregory Hickok. 2020. The Cortical Organization of Syntax. *Cerebral Cortex*, 30(3):1481–1498.

Mohammad Momenian, Zhengwu Ma, Shuyi Wu, Chengcheng Wang, Jonathan Brennan, John Hale, Lars Meyer, and Jixing Li. 2024. Le petit prince hong kong (lpphk): Naturalistic fmri and eeg data from older cantonese speakers. *Scientific data*, 11(1):992.

Christophe Pallier, Anne-Dominique Devauchelle, and Stanislas Dehaene. 2011. Cortical representation of the constituent structure of sentences. *Proceedings of the National Academy of Sciences*, 108(6):2522–2527.

Karalyn Patterson, Peter J. Nestor, and Timothy T. Rogers. 2007. Where do you know what you know? The representation of semantic knowledge in the human brain. *Nature Reviews Neuroscience*, 8(12):976–987.

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

Matthew A. Lambon Ralph, Elizabeth Jefferies, Karalyn Patterson, and Timothy T. Rogers. 2017. The neural and computational bases of semantic cognition. *Nature Reviews Neuroscience*, 18(1):42–55.

Alexander M Rapp, Dirk T Leube, Michael Erb, Wolfgang Grodd, and Tilo T. J Kircher. 2004. Neural correlates of metaphor processing. *Cognitive Brain Research*, 20(3):395–402.

Alexander M. Rapp, Dorothee E. Mutschler, and Michael Erb. 2012. Where in the brain is nonliteral language? A coordinate-based meta-analysis of functional magnetic resonance imaging studies. *NeuroImage*, 63(1):600–610.

Martin Schrimpf, Idan Asher Blank, Greta Tuckute, Carina Kauf, Eghbal A. Hosseini, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2021. The neural architecture of language: Integrative modeling converges on predictive processing. *Proceedings of the National Academy of Sciences*, 118(45):e2105646118.

William W. Seeley, Vinod Menon, Alan F. Schatzberg, Jennifer Keller, Gary H. Glover, Heather Kenna, Allan L. Reiss, and Michael D. Greicius. 2007. Dissociable intrinsic connectivity networks for salience processing and executive control. *Journal of Neuroscience*, 27(9).

Mariya Toneva, Tom M. Mitchell, and Leila Wehbe. 2022. Combining computational controls with natural text reveals aspects of meaning composition. *Nature Computational Science*, 2(11):745–757.

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

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. 2020. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3):261–272.

Qixuan Wang, Qian Zhou, Zhengwu Ma, Nan Wang, Tianyu Zhang, Yaoyao Fu, and Jixing Li. 2025. Le petit prince (lpp) multi-talker: Naturalistic 7 t fmri and eeg dataset. *Scientific data*, 12(1):829.

Shaoyun Yu, Chanyuan Gu, Kexin Huang, and Ping Li. 2024. Predicting the next sentence (not word) in large language models: What model-brain alignment tells us about discourse comprehension. *Science Advances*, 10(21):eadn7744.

Monika-Zita Zempleni, Remco Renken, John C. J. Hoeks, Johannes M. Hoogduin, and Laurie A. Stowe. 2007. Semantic ambiguity processing in sentence context: Evidence from event-related fMRI. *NeuroImage*, 34(3):1270–1279.

Rolf A. Zwaan and Gabriel A. Radvansky. 1998. Situation models in language comprehension and memory. *Psychological Bulletin*, 123(2):162–185.

Appendix

A fMRI data acquisition and preprocessing

MRI data were acquired with a 3 T GE Discovery MR750 scanner using a 32-channel head coil. Structural scans were acquired using a T1-weighted magnetization-prepared rapid gradient-echo sequence. Functional scans were obtained using a multi-echo echo-planar imaging (EPI) sequence (TR = 2000 ms; TEs = $[12.8, 27.5, 43]$ ms; flip angle = 77°; matrix size = 72×72; FoV = 240.0×240.0 mm; in-plane acceleration factor = 2; 33 axial slices; voxel size = $3.75 \times 3.75 \times 3.8$ mm). Each scanning run began with a trigger followed by an 8 s delay before stimulus onset. Anatomical and functional MRI data were preprocessed with fMRIPrep (v25.0.0; Esteban et al., 2020) with all default parameters, with final resampling to the "fsaverage4" surface performed in a single interpolation step using the `mri_vol2surf` function.

B Computational resources

All experiments were conducted on a high-performance computing (HPC) cluster with nodes equipped with dual AMD EPYC 7742 processors (64 cores per socket; 128 physical cores per node) and 512 GB of RAM. For each participant, the GLM required approximately 0.5 CPU-hours, and the banded ridge regression required approximately 5 CPU-hours.

C Banded ridge methods

In the banded ridge regression, fMRI responses y were predicted as

$$\hat{y} = \sum_i K_i w_i \tag{1}$$

where $K_i = X_i X_i^\top$ and w_i are the corresponding kernel weights. Model fitting minimized the objective

$$\|y - \sum_i K_i w_i\|^2 + \sum_i \alpha_i \, w_i^\top K_i w_i \tag{2}$$

with independent ridge penalties α_i for each kernel. We used the `precomputed` kernel option and performed random search over $\alpha_i \in [10^0, 10^{20}]$, optimizing log-weights $\delta_i = -\log \alpha_i$ via cross-validation. Data were split into the first eight runs for training and the final run for testing. Per-kernel predictions $\hat{y}_i$ were obtained using `predict(split=True)`, and Pearson correlations between predicted and observed responses were computed as model-specific performance scores.

Probing the Attention Representation of Filler-Gap Dependency in Transformers

Ruoqing Yao and **Pranav Anand**
Department of Linguistics, University of California, Santa Cruz
{ryao10|panand}@ucsc.edu

1 Introduction

Filler-gap dependencies (FGDs) require integration across unbounded intervening material, posing an important test case for the context-sensitivity of language models (LMs). Leveraging surprisal at the gap site, prior work has shown that RNN, LSTM, and transformer LMs show *unlicensed-gap* and *filled-gap effects* for FGDs within a clause (Wilcox et al., 2024; Kobzeva et al., 2025; Chang et al., 2025). As the dependency crosses multiple clausal layers, both effects attenuate (Wilcox et al., 2024; Kobzeva et al., 2025), an effect magnified with overt *that* intervening Cs (complementizers) (Kobzeva et al., 2025). Moreover, causal intervention experiments have shown that the internal representations in LMs for different categories of FGDs show commonalities that are modulated by structural frequency of the FGD category and filler type (Boguraev et al., 2025). The current study investigate how attention probing and ablation shed light on these results. We identify heads indexing FGDs near the gap site, explore how attention shifts in those heads with intervening material, find correlation of attention shift and surprisal results, and test head ablation's effect on surprisal.

2 Methodology

Following Wilcox et al. (2024) and Kobzeva et al. (2025), our items involve embedded *wh*-clauses, with 0C, 1C, or 2C clausal boundaries between the *wh*-FILLER and a a VERB object gap:

(1) I know FILLER . . . [(that) [. . . VERB __ . . .

We examine the attention matrices in the forward pass of GPT-2, which has 12 layers, each of which contains 12 heads. To quantify the attention from the VERB to the FILLER, from the full attention tensor, we extract the 12×12 attention weight where the query corresponds to the VERB token and the key corresponds to the FILLER *wh*-token.

3 Experiments

3.1 Experiment 1: Head Identification

In **Experiment 1a**, we created 15 sets of items as in (2a-b) to locate any heads that selectively attend from VERB to a FILLER. We also examined relative clauses (2c) to ensure that attention was not due to lexical differences of *wh*-words and *that*.

(2) a. I know {what, *that} the lion devoured __ yesterday.
b. I know {*what, that} the lion devoured the rabbit yesterday.
c. I know the rabbit that the lion devoured __ yesterday.

We identified the layer 5 head 2 (denoted $h_{(5,2)}$) and the layer 8 head 9 (denoted $h_{(8,9)}$) as selectively indexing FGDs: they both assign significantly greater attention from VERB to FILLER in grammatical FGD sentences (2a), relative to sentences without a gap (2b) (Figure 1a). As relative clauses show a similar attention profile, this indexation by $h_{(5,2)}$ and $h_{(8,9)}$ appears non-lexical. $h_{(5,2)}$ shows the greatest attention score among all 144 heads in [+*wh*] and [+RC], and $h_{(8,9)}$ has the second best score in [+*wh*] and shows the biggest difference between [+*wh*] and [+*th*]. A linear regression shows a significant positive correlation between the attention at both heads and the *filled-gap* surprisals ($\beta_{(5,2)} = 6.33$ and $\beta_{(8,9)} = 5.61$ respectively, $p_{(5,2)}, p_{(8,9)} < .001$).

Experiment 1b investigates how $h_{(5,2)}$ and $h_{(8,9)}$'s attentions change as linear or clausal structure intervenes between VERB and FILLER. We created 20 item sets, containing a) one 1C item with one clause intervening and three 0C items with a b) conjunction, c) mid-length modifier,or d) long-length modifier. Results show that $h_{(5,2)}$ attends to FILLER least in 1C items, and that all 0C

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 258–261
July 3-4, 2026 ©2026 Association for Computational Linguistics

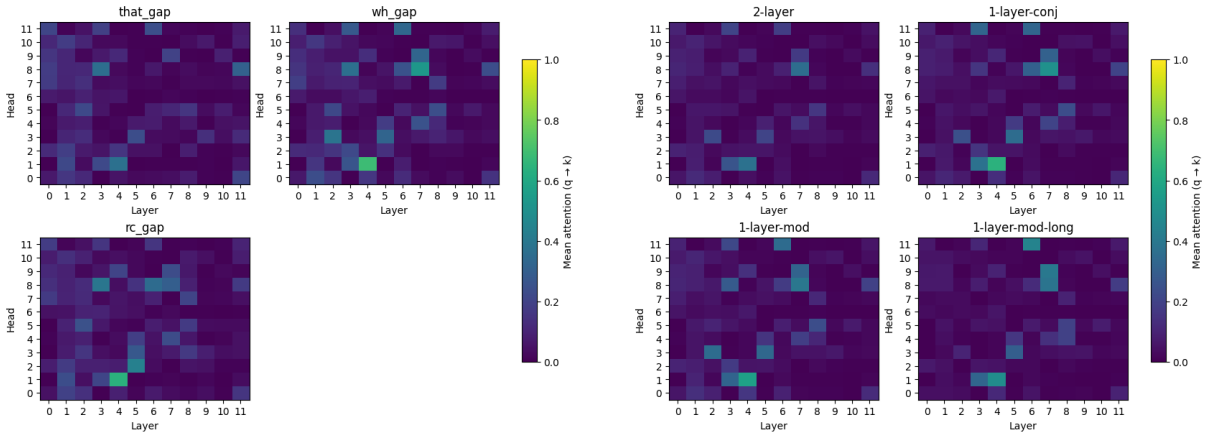


Figure 1: Experiment 1a (left) and 1b (right) results.

items show similarly high attention to FILLER (Figure 1b). This profile suggests that $h_{(5,2)}$ and $h_{(8,9)}$ indexes FGDs, independent of linear separation but sensitive to clausal intervention.

Experiment 1c tests whether the attention is attenuated by islands, following (Wilcox et al., 2024)'s surprisal finding. We adopted 50 sets of items from Kobzeva et al. (2025). Each set contains three conditions: a whether-island, a CNPC island, and a grammatical non-island baselineas in (3).

(3) a. *whether-island*: *I know what my brother confirmed whether our aunt brought ...

b. *CNPC island*: *I know what my brother confirmed the claim that our aunt brought ...

c. *non-island*: I know what my brother confirmed that our aunt brought ...

We found $h_{(5,2)}$ has higher attention across the conditions compared to $h_{(8,9)}$ (Figure 2). Both $h_{(5,2)}$ and $h_{(8,9)}$ show a contrast between non-island and CNPC islands, but only $h_{(8,9)}$ show the same contrast for *whether*-islands. $h_{(5,2)}$'s attention is relative to the type of islands – its attention is more attenuated in CNPC island, a strong island, than *whether*-island, a weak island. On the other hand, $h_{(8,9)}$ indexes islands categorically as a whole regardless of the island types.

3.2 Experiment 2: Overt Complementizers

A puzzling finding in Kobzeva et al. (2025) is that overt (*that*) *C* neutralize unlicensed- and filled-gap effects. We observed above that $h_{(5,2)}$ and $h_{(8,9)}$

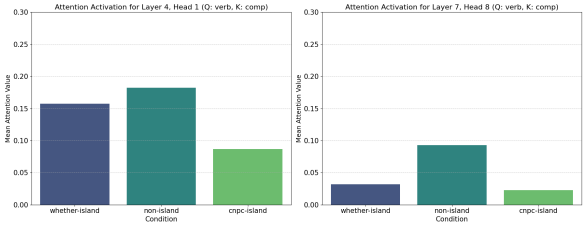


Figure 2: Experiment 1c Results

attends to FILLER, and **Experiment 2** tests whether this can explain the neutralization in surprisal with overt *that C*'s. We adapt 20 1C and 2C item sets from Kobzeva et al. (2025), varying for 2C items whether BOTH *C*'s, only the HIGH one, only the LOW one are overt, or NONE. Items are schematized in (4).

(4) a. 1C: She knows what he heard $[_{CP_{loc}}$ (that) the priest revealed at the party.

b. 2C: She knows what the newspaper reported $[_{CP_{int}}$ (that) he heard $[_{CP_{loc}}$ (that) the priest revealed at the party.

Overall, we observe an *"attend closest C"* pattern of $h_{(5,2)}$ (Figure 3). A linear regression reveals that $h_{(5,2)}$'s attention to the FILLER significantly correlates with filled-gap surprisal ($\beta = 12.59$, $p < .001$). In 1C cases, when the local C_{loc} is overt, $h_{(5,2)}$ attends at VERB principally to C_{loc}, and attention to FILLER is substantially reduced when compared to non-overt C_{loc}. In 2C cases, with no overt *C*s, $h_{(5,2)}$ at VERB selectively attends to FILLER, though the magnitude reduces relative to 1C cases, and it further diminishes with overt *C*s. Noticeably, C_{loc} is principally attended to by $h_{(5,2)}$ whenever overt to consistent magnitudes similar to

that in 1C. When BOTH intervening *C*s are overt, FILLER and C_{int} are similarly attended to well below C_{loc}. With only C_{int} overt, its magnitude still surpasses FILLER, though to a smaller extent. We note a similar *"attend closest C"* pattern in relative clauses as in (2c): VERB attends to the overt C more than the NP FILLER.

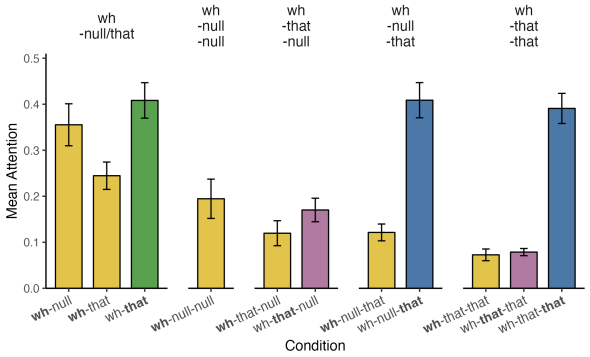


Figure 3: Experiment 2 Results on $h_{(5,2)}$

$h_{(8,9)}$ shows a different pattern (Figure 4). Except the *wh-that-that* condition, it assigns the most attention to the FILLER *wh*-word. In the 1C condition, having C_{loc} reduced its attention to FILLER, but its attention to FILLER is still significantly higher than C_{loc}. The 2C condition shows a similar pattern: including overt C_{int} and C_{loc} reduces the attention to FILLER, but the FILLER attention remains the highest. $h_{(8,9)}$'s attention to FILLER is reduced by the inclusion of overt local *C*s, yet its attention does not redistribute.

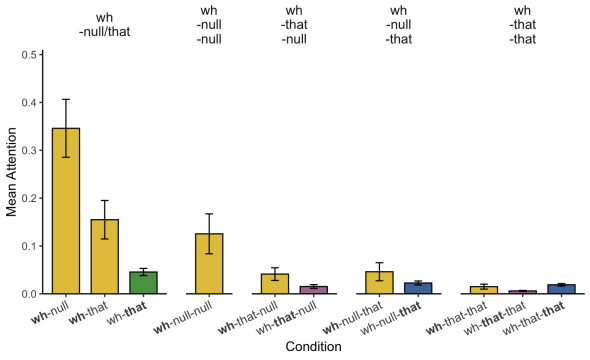


Figure 4: Experiment 2 Results on $h_{(8,9)}$

3.3 Experiment 3: Head Ablation

We conduct a one-head-at-a-time ablation study in GPT-2. Using TransformerLens (Nanda and Bloom, 2022), we zero out each attention head during the all forward pass blocks and measure the resulting change in surprisal. We measure the surprisal difference between the ablated model and the full model for both [+gap] and [-gap] across three conditions [+*wh*], [+*th*], and [+RC] in (2). A [+*wh*] example is repeated in (5) with the measured target underlined. If a head contributes to FGD computation, its ablation should weaken the model's sensitivity to the relevant contrast. For ungrammatical targets, the ablated run of a FGD-correlated head should produce lower surprisal than the full model because the high surprisals of the ungrammatical continuations should be attenuated, and the reverse for grammatical targets.

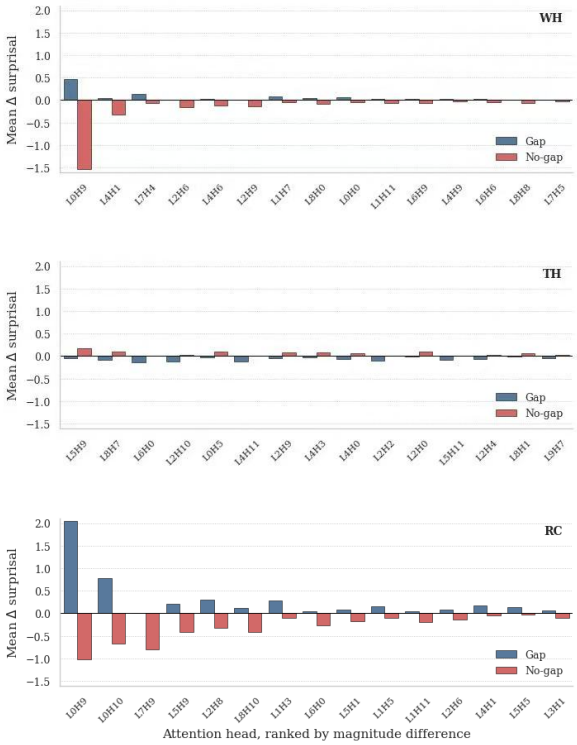


Figure 5: Ranked mean ablation effects for top 15 attention heads for [+*wh*], [+*th*], and [+RC] conditions. Heads are ordered by the magnitude differences.

(5) a. I know what the lion devoured __ yesterday ...

b. *I know what the lion devoured the rabbit ...

The results (Figure 5) show that a small subset of attention heads yields the predicted surprisal differences rather than an uniformly distributed effect across the network. The strongest ablation effect is found for $h_{(1,10)}$, a head that does not show strong VERB-to-FILLER attention in the probing analyses, suggesting that some causally relevant heads may encode the dependency independent of our assumed VERB-to-FILLER attention. Crucially, $h_{(5,2)}$,

one of the heads identified in the attention-probing experiments, produces the second strongest ablation effect in [+*wh*] and mild effects in [+RC], indicating that its attention correlation with FGD extends to ablation. By contrast, $h_{(8,9)}$ did not produce any of the predicted surprisal changes under ablation. These results suggest that GPT-2's FGD behavior is supported by a partially specialized but distributed set of heads, and that attention weights alone provide a meaningful but incomplete picture of the model-internal mechanisms underlying syntactic dependency processing.

[+*th*] show a different set of effective heads, with much smaller ablation effects than [+*wh*] and [+RC]. This is expected, since [+*th*] lacks an active filler. The contrast suggests that GPT-2's dependency computations are more engaged when a potential filler is present.

4 Discussion

Our results of $h_{(5,2)}$ provide a potential explanation for Kobzeva et al.'s (2025) observation that surprisal effects attenuate with overt *C*s: $h_{(5,2)}$ redistributes its attention from FILLER to the most local C. This might indicate that the transformer has a less reliable representation of FGDs across overt *C*s, and hence is less able to distinguish whether there are FGDs or not. If we take this correlation seriously, it suggests that unbiased language learners, such as LMs, fail to fully learn correct grammars of FGDs, which should not be sensitive the overtness of C heads (Ritchart et al., 2016).

Why might this be? One possibility is GPT2's training data, in which 1C structures are likely vastly more common than 2C ones, particularly relative clauses with C_{loc}, where FILLER and C_{loc} are both reliable indicators of FGDs. The model might overfit to attend to C_{loc} whenever it's present, reflecting $h_{(5,2)}$ attention to both FILLERs and Cs. Relevant here is Boguraev et al.'s (2025) investigation of representational difference across FGD type (matrix and 1C embedded questions), RCs, (pseudo)clefts, and topicalization): representations for a source FGD generalize to the others, in correlation with source frequency. However, differences in token order seem to matter (RCs generalize surprisingly poorly, despite being the most prevalent construction), as do embedding (matrix questions generalize little to embedded questions). Similarly, we might wonder if 1C cases generalize to 2C cases in GPT-2, and how the presence of overt *that* interacts with generalization.

We observed a functional distinction of $h_{(8,9)}$ from $h_{(5,2)}$. It is attenuated but not attracted by C_{int} and C_{loc}, sensitive to islands categorically, and not effective when ablated. These findings point to the possibility that LMs have allocated distinct linguistically meaningful representations from the training data to individual attention heads. They may each track FGDs in a unique way, and may or may not collaborate to represent FGDs. It is, admittedly, puzzling why ablating $h_{(8,9)}$ does not produce any effect, and why ablating $h_{(1,10)}$ is highly effective without attending from VERB to FILLER. Both of these we leave to future investigation.

Despite our findings reported here, we do not claim that $h_{(5,2)}$ and $h_{(8,9)}$ have causal effects on FGD performance in LMs, as attention probing and ablation are both indirect, correlational approximation of the FGD phenomena in the LMs. For attention probing, a head attending strongly from VERB to FILLER does not necessarily mean the head is *computing* the filler-gap dependency, so that any effects found in attention probing is merely a model-internal correlation to a given pattern. While ablation investigates effect, it does so by rather strongly altering the learned network, and its destructive effect to the model's computation makes it hard to interpret cleanly. We leave further experiments with causal intervention methods such as activation patching to future research.

References

S. Boguraev, C. Potts, and K. Mahowald. 2025. Causal interventions reveal shared structure across english filler–gap constructions. In *Proceedings of the 2025 EMNLP*, pages 25032–25053.

C.-Y. Chang, X. Huang, H. Nasir, S. Storks, O. Akingbade, and H. Dai. 2025. Mind the gap. In *EMNLP 2025*, pages 15060–15076.

A. Kobzeva, S. Arehalli, T. Linzen, and D. Kush. 2025. Learning filler-gap dependencies with neural language models. *JML*, 144:104663.

N. Nanda and J. Bloom. 2022. Transformerlens. https://github.com/TransformerLensOrg/TransformerLens.

A. Ritchart, G. Goodall, and M. Garellek. 2016. Prosody and the that-trace effect. In *WCCFL 33*, pages 320–328. Cascadilla Proceedings Project.

E. Wilcox, R. Futrell, and R. Levy. 2024. Using computational models to test syntactic learnability. *LI*, 55(4):805–848.

Learning Stress in Arabic Low-Resource Settings

Abdelrahim 'Abed' Qaddoumi[1] Salam Khalifa[1,2] Jordan Kodner[1]
Jeffrey Heinz[1] Owen Rambow[1]
[1]Institute for Advanced Computational Science and Department of Linguistics,
Stony Brook University
[2]Computational Approaches to Modeling Language (CAMeL) Lab, NYU Abu Dhabi
{first.last}@stonybrook.edu

Abstract

We predict lexical stress in Arabic varieties from syllable structure, modeling stress assignment as generation: given an unstressed input, the system outputs a stress-marked word. We compare four approaches: a grammar induction algorithm (BUFIA), a transformer-based neural network, a rule-based method derived from linguistic literature, and a frequency baseline. The models are evaluated across several low-resource settings by varying the training data size by words, structural type, and syllable count. BUFIA outperforms the neural network, especially when data are scarce. This points to grammar induction as an interpretable and sample-efficient approach for learning stress.

1 Introduction

Lexical stress is the emphasis of one syllable in a word. Stress may be contrastive, as in many American English verb-noun pairs (recórd (verb) vs. récord (noun), prodúce (verb) vs. próduce (noun)), or may be completely predicted from phonological structure, as is the case for the Arabic varieties discussed in this paper. Closely related language varieties may vary in their stress patterns, as is the case for Arabic, and for English as well. For example, British English may the stress *weekend* second syllable, whereas American English stresses the first. Southern varieties of American English may employ initial stress in words like *police*, where other varieties stress the final syllable.

In addition to its roles in linguistics *per se*, stress can enhance engineering applications like automatic speech recognition and text-to-speech systems (Wang and Seneff; Ludusan et al., 2012; Zou et al., 2021; Geneva et al., 2023). Building on these applications, research sometimess focuses on automatic stress prediction Dou et al. (2009); van Esch et al. (2016).

In linguistics, stress in Arabic has arguably received the most attention among languages outside English (Watson, 2011). This attention is due to Arabic varieties providing an excellent environment for investigating stress. All varieties exhibit stress and share similarities such that stress location is a function of both syllable shape and position. Yet, they differ in the distribution of syllable types, the leftmost extent of stress (third or fourth syllable from the right),[1] and the degree to which lexical information beyond phonology may affect stress (Watson, 2011).

Computationally, Arabic presents an interesting case because of its diglossia (i.e., the use of two varieties, one for spoken and another for written). Modern Standard Arabic (MSA), the written variety, has good resources, while most spoken varieties have few. This resource imbalance creates a challenge for practical modeling in data-scarce settings. However, since Arabic regional varieties are related and vary in predictable ways, they provide ample ground for cross-variety transfer techniques. Still, computational research for Arabic stress has been limited.

In this paper, we study stress prediction in three Arabic varieties: Egyptian, Sudanese, and Jordanian. As Arabic stress is strongly constrained by syllable structure and is determined within a limited window near the right edge of the word, we represent each word as a sequence of CV syllables (e.g., <كاتب> [ˈkaa.tib] 'writer' is represented as `cvv.cvc`). This allows us to generate the stressed version of the word by capitalizing the stressed syllable (i.e., `CVV.cvc`).

We compare four approaches: (1) BUFIA is a grammar induction approach that learns constraints and compiles them into a finite-state acceptor/transducer (Chandlee et al., 2019), (2) Neural transducer, a transformer-based character-level

[1]We use "left" to refer to the start of words and "right" to reference the end, keeping in line with most literature on stress. Our data consists of left-to-right IPA transcriptions rather than right-to-left Arabic script.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 262–279
July 3-4, 2026 ©2026 Association for Computational Linguistics

encoder–decoder (Wu et al., 2021), (3) a strong baseline rule-based implementation derived from the literature in descriptive linguistics on each variety's stress system, and (4) Majority baseline predicting the most frequent stress position.

We evaluate all approaches in low-resource conditions by varying the size of the training data across three distinct measures: random word-based sampling, structural-type sampling (where the training set is constructed by progressively including larger proportions of syllable-structure types available in the language), and length-based sampling (where words of increasing syllable count expand the training dataset). This evaluation design enables us to test whether a model can generalize to unseen structural types and longer words, moving beyond a limited number of words encountered during training.

Our results indicate that BUFIA is a viable modeling strategy for phonological generalizations when data are limited, and interpretability is desirable. The paper makes the following contributions: it creates stress-marked datasets for Jordanian and Sudanese, compares BUFIA and a neural transducer in multiple low-resource settings, and develops a method for testing generalization using linguistic data.

The remainder of the paper is organized as follows: §2 reviews relevant linguistic background; §3 describes the grammar induction algorithm (BUFIA); §4 presents the datasets; §5 and §6 detail the experimental setup; §7 reports results; and §8 provides error analyses and discussion.

2 Linguistic Background

2.1 Computational Stress Learning

Computational learning of stress has a long history in linguistics. Dresher and Kaye (1990) frame learning stress as a setting in which a small number of metrical parameters are set in response to phonological cues. Gupta and Touretzky (1991) explore whether a perceptron can be trained to predict stress, demonstrating one of the earliest statistical approaches to learning stress. Goldsmith (1994) approach the study of accent systems using dynamic computations. In the early 2000s, with the rise of Optimality theory, researchers studied the problem through the lens of constraints and their rankings (e.g., (Tesar and Smolensky, 2000)).

Hayes and Wilson (2008) uses weighted constraints in a Maximum Entropy (MaxEnt) phonotactic learner to learn stress, among other phonological phenomena. Heinz (2009) argue that using finite-state machines and simple learners that attest to stress systems can be characterized by local properties and learn from their local patterns. Recently, many studies have examined stress using statistical approaches such as neural networks (Prickett and Pater, 2025) and MaxEnt (Lee et al., 2025). While our work shares the same problem as we are trying to learn stress, we differ in the goal of the research, as we are trying to use stress as an application to put a lens on the difference between two modeling approaches, one being neural networks and the second being grammar induction algorithms.

2.2 Arabic Varieties

While often treated as a single language, modern Arabic is actually a collection of regional varieties that differ in their phonological patterns. Our study focuses on three varieties: the Ammani variety of Jordanian Arabic, the Khartoum variety of Sudanese Arabic, and the Cairene variety of Egyptian Arabic. While this choice was constrained by data availability, the three varieties provide good coverage of Arabic stress patterns. We use the existing resources when available for the varieties described in Section 4. Data was checked by the first and second authors, who are native speakers of Ammani and Sudanese respectively.

Jordanian Arabic (JOR), spoken by around 12 million people in Jordan (World Bank, 2024), falls within the higher level grouping of Levantine varieties (Eberhard et al., 2025). Jordanian, in turn, is comprised of sedentary and Bedouin varieties (Palva, 1984). The former may be further divided into Ammani (spoken in the capital region), Northern Jordanian, Southern Jordanian, and Aqaba (Herin et al., 2021).

Within Egyptian Arabic (EGY), which has about 83 million speakers (Eberhard et al., 2025), we focus on Cairene, the urban variety of the national capital. This variety, with approximately 20 million native speakers, is unusual among regional varieties for its prominence in the media, which makes it widely understood across the Arab world (Khalil, 2011).

Sudanese Arabic is spoken by about 41 million people in Sudan, South Sudan, and Eritrea (Eberhard et al., 2025). It is usually grouped with Egyptian Arabic in the Egypto-Sudanic or Nile Valley category (Versteegh, 2014). According to Gasim

(1965), there is a difference between the sedentary dialects along the Nile and the pastoralist dialects in western Sudan. Most phonological research focuses on the prestige variety, known as Khartoum Arabic or Sudanese Standard Arabic (Mustapha, 1982; Manfredi, 2015).

2.3 Stress and Syllables

Stress is an abstraction that captures a constellation of distinct physiological phenomena. Different languages employ different combinations of these phenomena to indicate stress, including but not limited to loudness, pitch, and duration (Ladefoged and Johnson, 2014).

Stress predictability differs by language: it can be lexically specified, in which case speakers memorize the stress pattern for each word, or phonologically predictable from syllable weight or position. It can also be a mixture of lexical and phonological features. Finnish is a language with phonological stress (Suomi et al., 2009), whereas Tagalog has contrastive lexical stress (e.g., words differ only in stress) (Schachter and Otanes, 1983), and according to Vendelin (2010) Spanish stress is a hybrid pattern as it is lexically contrastive but constrained to a right-edge window with a preference for penultimate stress (Gordon and van der Hulst, 2020).

In systems with predictable stress, stress can be assigned mechanically (Hayes, 1995). Arabic varieties' stress patterns are phonological, although they differ in the details of stress assignment. Stress placement is usually computable by looking at the heaviest syllable from the rightmost edge of the word (assuming a left-to-right transcription for simplicity) (Watson, 2002; Broselow, 2017).

Stress may be mechanically assigned to specific syllables in a word according to their syllable weight. To explain this, we provide a brief review of syllable structure. See (Goldsmith, 2011) for a more thorough introduction. We assume the standard decomposition of a syllable into onset-nucleus-coda. Syllables must contain a nucleus, which is usually a vowel (`v`), and they may include some number of consonants (`c`) in the onset before the nucleus and/or in the coda after the nucleus. A long vowel is represented by `vv`.

Languages differ in which combinations they permit. All Arabic varieties require that each syllable contain an onset, so the minimal licit syllable is `cv` (Broselow, 2017). The number of segments in a syllable's nucleus and coda determines its weight (Broselow et al., 1997). Syllables with only a short vowel in the nucleus and a `c` in the onset, `cv`, are light, those with a long vowel or a short vowel and a single coda consonant are heavy (`cvv`, `cvc`), and those with a long vowel and coda or multiple coda consonants are super-heavy (`cvvc`, `cvcc`, `cvvcc`, etc.). Some varieties may treat word-final syllables differently from other positions, for example, treating a word-final `cvc` syllable as light (Hayes, 1995).[2]

There are two further complexities of stress which our system does not account for. First, our representations for Arabic do not include secondary stress. It has been argued that Arabic lacks secondary stress (Al-Jarrah, 2002), and it is not included in our datasets, but see Becker (2022). Second, stress may interact with phrase- and sentence-level prosodic patterns, but this is not applicable to our model, which runs on word lists.

3 BUFIA

This section describes the Bottom Up Factor Inference Algorithm (BUFIA; Chandlee et al., 2019; Rawski, 2021; Swanson et al., 2026). This algorithm identifies a finite set of constraints that are satisfied by the training data.

Words are represented as mathematical structures containing substructures (Rogers and Lambert, 2019). Words are well-formed only if all of the substructures they contain are well-formed. Conversely, the presence of any ill-formed substructure in a word means the word itself is ill-formed. Consequently, a constraint is just an ill-formed structure. For example, consider the formal language $0(ba)^*0$, which represents words as strings of a, b with word boundaries explicitly marked with 0. A grammar for this language can be given by the finite set of ill-formed structures $0a, aa, bb, b0$. As constraints, these structures pick out the language whose words do not begin with a, do not contain aa substrings, do not contain bb substrings, and do not end with b.

BUFIA works with any representational schema for words, provided the schema is fixed in advance. For instance, instead of representing b as an atomic symbol, it could be defined as a vector of feature values such as [+stress, +consonant, −vowel]. In this way, a constraint banning adja-

[2] For more information about the allowed syllable structure, see Appendix Table 8, which lists each variety's licit syllable structures sorted by their weights.

cent stress in words could be given as the substructure [+stress][+stress].

BUFIA is designed to find the structures *absent* in data. It can do this because the structure containment relation partially orders the space of logically possible constraints. It conducts a breadth-first traversal of this space from the bottom up and checks whether the structures it encounters are present in the training data. If not, it adds them to the grammar it eventually outputs. It then also knows that all structural constraints that contain the found one need not be checked. BUFIA proceeds in this way until a stopping condition has been met. In this paper, the stopping condition was set to limit the length to a maximum of four segments. Chandlee et al. (2019) proves that BUFIA returns a grammar whose language is (1) the smallest one in the class of languages consistent with the data, and (2) contains the most general constraints of any other grammar in the class satisfying (1).

Appendix Table 6 shows an example of the constraints for one Jordanian BUFIA result. The table is split into different sections. For example, we have [][+wb] and [+wb][], which means that word boundaries only occur at word edges. Then, we have [+vowel][+vowel][+vowel], meaning there can not be three vowels in one sequence.

4 Data

The Egyptian data comes from the Egyptian Colloquial Arabic Lexicon (Kilany et al., 2002). The lexicon is drawn from 140 CALLHOME telephone conversations of native speakers of Cairene Egyptian Arabic. Each of its 51,202 data points provides a surface (spoken) form, its length in syllables, and the stress pattern indicating which syllable is stressed. Since the actual syllabification is not provided, use Simple Syllabify tool (Kodner, 2016), which identifies syllable nuclei and then inserts syllable boundaries following the Maximum Onset Principle (Goldsmith, 2011). The syllabifier, which we evaluate in (Qaddoumi et al., 2026), performed well across several Arabic varieties, achieving accuracies of 99% on EGY, 98% on JOR, and 97%+ on other available varieties.

We collected the Jordanian data by scraping all entries for South Levantine Arabic on Wiktionary.[3] This initially yielded around 2,782 entries, which included IPA transcriptions indicating both syllabification and stress, which we then filtered through deduplication, which ultimately left us with 2,082 words.

We collected the Sudanese data by extracting all the examples used in (Hamid, 1984). The data was collected and verified by the second author. There were 756 entries in total after we removed phrases that have two stresses or more, since the rules of a phrase's stress are different from lexical stress. In addition to this, we remove any words missing stress annotation.

4.1 Preprocessing

We preprocess the data to remove errors and inconsistencies introduced by the original annotation and to prepare it for BUFIA and the neural network. Wiktionary in particular, while an overall reliable source, is crowdsourced, and is subject to a degree of noise (Sakunkoo and Sakunkoo, 2025). We remove duplicates and entries with transcriptions containing only one or two characters, since they are trivial. We further standardize transcriptions as follows: Long vowels are represented as a repeated character, for example, <كاتب> [ka:.tib] 'writer', will be [kaa.tib]. Geminates are represented by repeating the same character, for example, <كتّب> [kat:ab] 'he made x write', will be represented as [kat.tab].

When an entry provides multiple IPA transcriptions, we retain each distinct transcription as a separate surface form for that variety, for example, <شُوكُولَاتَة> [ʃu.kuu.laa.ta] or [ʃu.ku.laa.ta] 'chocolate'. For the cases where one segment is presented as optional in a single transcription, we always include the optional segment, as in the following example, where <اسمر> /(ɪ)smar/ 'dark' is treated as [ɪs.mar].

All superscripts such as 'ˤ' and subscripts such as '̪' are removed as well, since they are not themselves segments and they correspond to annotations or to phonetic distinctions which do not affect Arabic stress (Association, 1999), for example, the tie bar '͡' and pharyngeal diacritics 'ˤ' in <آية الله> /ʔaa.ja.tu͡lˤ.lˤaah/ 'sign of God'. We also remove [ ] and / /.

5 Methodology: Problem Formulations

General Problem Formulation Our task is to predict which syllable in a word receives primary stress. The input is a phonetic transcription where the vowels are automatically replaced to `v` and the consonants to `c`. The usage of CV sequences is mo-

[3] https://www.wiktionary.org/

tivated by the fact that stress assignment in Arabic and other languages is mainly governed by syllable weight, which is determined by syllable structure in terms of vowels and consonants rather than the particular identities of these segments.

We define the problem as: given a word **x** consisting of **n** syllables each represented as a sequence of characters `c` (consonants) and `v` (vowels), a **generator** should output the word with the stressed syllable marked. We adopt the convention of indicating the stressed syllable with upper case `C` and `V`. For example, given the input $\mathbf{x}_g =$ `c v c v v c`, the output should be $\mathbf{y} =$ `c v C V V C`.

To test the capabilities of BUFIA and NN, we evaluate the models under three sampling paradigms. Within each same random seed and the specific sampling paradigms the models are evaluated on identical splits. The paradigms differ in how the training, dev, and test set are constructed and each is designed to test a different kind of generalization, thus it will give us a deeper insight into the nature of generalization. However, since the dev and test sets differ for the three sampling paradigms, the results from different sampling paradigms cannot be compared. Indeed, we are not trying to compare the three paradigms; rather, we are using them to perform different comparisons of the two learning algorithms we are investigating (BUFIA and neural learning).

5.1 Data Splits

Across all sampling paradigms we split the data described in Section 4 following a 80/10/10 train/dev/test split. Then we test both models on the full size of the dev and test datasets. These splits are done 5 times with different seeds.

Details about training set sizes for all sampling strategies are provided in Table 9 and Appendix Tables 10, 11, 12, 13, 14, and 15.

Sampling Paradigm 1: Random word sampling (data-quantity scaling). This paradigm measures how well models perform as we scale the training data quantity. The data is randomly picked without any constraints. Test and dev sets are drawn first, at random, fixed for each seed, and each is 10% of the total data. We then construct seven *nested* training subsets at approximately 1% of the training data, then 3%, 6%, 13%, 25%, 50%, and finally 100% of the training partition. For exact numbers per dataset, check Table 9.

SC	#CVTs	Test OOV	Dev OOV
≤2	34	51.16%	41.67%
≤4	80	20.93%	16.67%

Table 1: The percentage of CV templates represented in the test and dev but not included in the train set for the sampling by syllable counts (Paradigm 2). The SC column indicates the syllable counts in training subset. The column #CVTs is the number of CV structures present in training set, while Test and Dev OOV indicate the percentage of CV structures not present in those splits which are absent from the training set. These numbers are for seed 21 for Jordanian just to illustrate the distribution of data.

Sampling Paradigm 2: Sampling words by their syllable count (generalization to unseen lengths). This sampling approach is sensitive to words' length syllables, in order to evaluate models' ability to generalize to unseen lengths.

To generate the splits for this experiment, words are sorted by their lengths in syllables. For example, the full JOR dataset contains 328 monosyllabic, 1,356 disyllabic, 382 trisyllabic words, and 15 words with four syllables. Due to the limited number of four-syllable words in Jordanian, we decided to group them with trisyllabic words.

Sampling then proceeds as follows: First, we ensure that the test and dev datasets sample all lengths by sampling one word of each length for both partitions. Second, sampling continues until the full dataset is exhausted, ensuring an approximately 80/10/10 split.

The training set is then subsampled to produce learning curves as for Paradigm 1. However, subsampling is sensitive to words' syllable count. We first sample all monosyllabic and disyllabic words. We then create a larger superset by sampling word trisyllabic words, and so on, until finally the entire training data is included. These subsamples create scenarios where dev and test contain words with syllable lengths unattested in training.

The training set is again split by syllable count. We start with sampling words made of 1 or 2 syllables counts. Then in the next step, we sample from 3 syllables increasing syllable counts until it reaches as many as we can obtain from the training data. Note that there might be syllable counts that are present in test and dev but that do not exist in the training. In Table1, we can see an example for the out-of-vocabulary distribution for seed 21 for Jordanian.

Size (%)	#CVTs	Test OOV	Dev OOV
1	7	93.75%	93.75%
3	7	93.75%	93.75%
6	8	92.5%	92.5%
13	13	87.5%	87.5%
25	20	81.25%	81.25%
50	42	60%	58.75%
100	58	41.25%	41.25%

Table 2: The percentage of CV templates represented in the test and dev but not included in the train set for the Sampling by CV structures. The size column represent the training subset size, #CVTs is the number of CV structures present in training set, and Test and Dev OOV represent the percentage of CV structures not present in training set. These numbers are for seed 21 for Jordanian just to illustrate the distribution of data.

Sampling Paradigm 3: Sampling words by their CV structures (generalization to unseen templates). This paradigm evaluates models' ability to generalize to unseen syllabic structures. We define a unique *structure type* as a pattern of `c` and `v` segments, ignoring stress. For example, `cv.cvc`, `cvc.cvc`, and `cvc.cvvc` are three distinct structure types. While `CV.cv.cv`, `cv.CV.cv`, and `cv.cv.CV` are one structure type.

To generate the data splits for this paradigm, we begin by sorting words by their structure types. For example, in JOR there are 271 words with the `cv.cvc` structure type, 265 with the `cvc.cvc` structure type, 129 with the `cvc.cvvc` structure type, and so on

Parallel to the sampling for Paradigm 2, we ensure that dev and test contain at least one instance of every structure type present in the data set before regular sampling. One caveat to this constraint is that a structure type that only occurs once in the full data set will only occur in dev or test for a given random seed, and not both.

The training data is once again subsampled, this time in a way that is sensitive to syllable types. We start by selecting a number of structure types that generate approximately 1% of the whole dataset, then 3%, and so on, until we reach the full size of the training dataset. This number varies; for example, we could sample 17 entries with 3 structure types in the first step, or 129 entries with 1 structure type. The number of samples depends on the 1% threshold applied to the entire dataset.

For example, if we have 1500 words in training dataset and we are sampling and we select a structure type with 129 entries, which is larger than 1% of the training dataset, we stop sampling and start creating the next split which is 3%. Now, since we already have 129 words in the training data then sampled structure type will stop the sampling and move to the next step, since 129 is larger than 45. We keep repeating this until we have the 7 steps. For the exact number of entries, and structure types, check Appendix Tables 10, 11, and 12

6 Methodology: Models and Baselines

Models and Baselines We evaluate two models on generation: BUFIA (Section 6.2) and a neural transducer (Section 6.1). We have two baselines, the majority baseline of picking the most frequent stress position in each language, and a rule-based implementation derived from the descriptive literature on each variety's stress system. The rule-based baseline is a serious baseline, as much scholarly work has addressed this issue (Section 6.3).

6.1 Neural Network (NN)

Our neural model is the character–level transformer presented in (Wu et al., 2021), which has been widely strong baseline for morphological inflection, historical text normalization, grapheme-to-phoneme conversion, and transliteration tasks. The model is a standard encoder–decoder with multi-head self-attention. This is trained using the syllable structure representation adopted throughout this paper (i.e., given input `c v c v v c`, the model should generate `c v c v v c`).The parameters of the model are described in Table 7.

6.2 BUFIA-Derived Finite State Machine

We used an off-the-shelf C++ implementation of BUFIA.[4] For easier processing, we transform the constraints generated by BUFIA into a constraint format suitable for using The Language Toolkit (LTK) (Lambert, 2024) to generate language-equivalent finite state acceptors (FSAs). The generated FSAs can be used directly to evaluate the accuracy of the induced grammar. The evaluation is done based on a generation task in which the FSA receives an unstressed input and outputs the stressed version.

The FSAs generated from LTK are unweighted. This causes an issue when there is apparent optionality in the data. For example, if the data has a list

[4] https://github.com/pterodactylogan/bufia

of 100 inputs of the form `cv.cv`, with 99 having stress pattern `CV.cv` and one `cv.CV`, then both inputs will be accepted because they are valid paths. We therefore add weights to the FSA to account for the distributions seen in the training set. The idea is to count the total number of transitions in a valid path using the Viterbi algorithm.

Once we have the total count for all segments present in a valid path in the training dataset, we start assigning weights state by state. We do this by going from the start state, find the outgoing arcs, evaluate the total count of outgoing arcs, and compute a smoothed probability using $p = (count + \alpha)/(total_count + \alpha * m)$, where $\alpha = 0.5$, and m is the number of outgoing arcs. Then we set the cost to $c = -log(p)$.

For the generator FST, the state machine generates all toggle versions and picks the lowest-cost path if there are multiple paths. This means the input `cv.cv.cv` generates three inputs `CV.cv.cv`, `cv.CV.cv`, and `cv.cv.CV`. Thus, each input generates multiple paths with different costs. The FSA "outputs" the path with the lowest cost and the other paths are invalid.

6.3 Linguistic Knowledge: Stress Assignment in JOR, SUD and EGY

This experiment implemented the rules mentioned for Jordanian, Sudanese, and Egyptian stress assignment.

We use linguistic literature to develop a sophisticated baseline in addition to the most frequent position baseline. To implement the sophisticated baseline and test it, we need an abstract word representation that captures only syllable weight and stress, since that is how linguists predict Arabic stress. Unstressed syllables are represented as *l* (light), *h* (heavy), and *x* (super-heavy). Stressed syllables are capitalized *L*, *H*, and *X*. For example, Jordanian Arabic مكاتب [ma.ˈkaa.tib] 'offices' contains the syllable structures `cv.CVV.cvc`, corresponding to the weight representation [*l H h*]. The data in Appendix Table 8 contains all the syllable structures present in our data and their weights.

In JOR, stress and vowel length together are contrastive, yielding near-minimal pairs like مكاتيب [ma.ka.ˈtiib] 'letters' and مكاتب [ma.ˈkaa.tib] 'offices' derived from the same root. JOR stress has been subject to several theoretical treatments in terms of traditional ordered rule phonology and metrical phonology (Al-Sughayer, 1990; AbuAbbas, 2003; Al-Wer, 2007). We implemented the following rules from the cited work:

(1) **Jordanian Stress Assignment**

- Stress the rightmost heavy or super-heavy syllable if it is final, penultimate, or antepenultimate.
- Stress the antepenultimate syllable otherwise.

Stress assignment in Cairene Egyptian Arabic is generally predictable and has been the subject of several treatments in different frameworks (Broselow, 1976; McCarthy, 1979; Watson, 2002). Following Becker (2022), we summarize stress assignment in 2. Stress is predictable if one of the last three syllables is heavy or the word consists of one to four light syllables.

(2) **Cairene Stress Assignment**

- If the final syllable is heavy or super-heavy, assign final stress.
- If the final syllable is light and the penult is heavy, assign penultimate stress.
- If the final two syllables are light and the antepenult is heavy, assign the penultimate stress.
- If the word is disyllabic and both syllables are light, assign penultimate stress.
- If the word is a monomorpheme, assign final stress.
- Else, stress the antepenult in trisyllabic or longer words.

Similar to JOR and EGY, stress in SUD is very predictable and has been studied by several researchers, including (Abdel-Khalig, 2014; Ali, 2017; Dickins, 2007). We implemented the slightly modified following rules based on (Hamid, 1984):

(3) **Sudanese Stress Assignment**

- Stress the heavy or super-heavy syllable if it is the only one.
- If there is more than one heavy or super-heavy syllable, stress the rightmost one.
- Otherwise, stress the first syllable

6.4 Baseline

We use a standard simple baseline based on the most frequent stress pattern. The most frequent stress index in Jordanian Arabic is the first syllable, with 72.7%. The most frequent stress index in Egyptian Arabic is the second syllable with 55.0%.

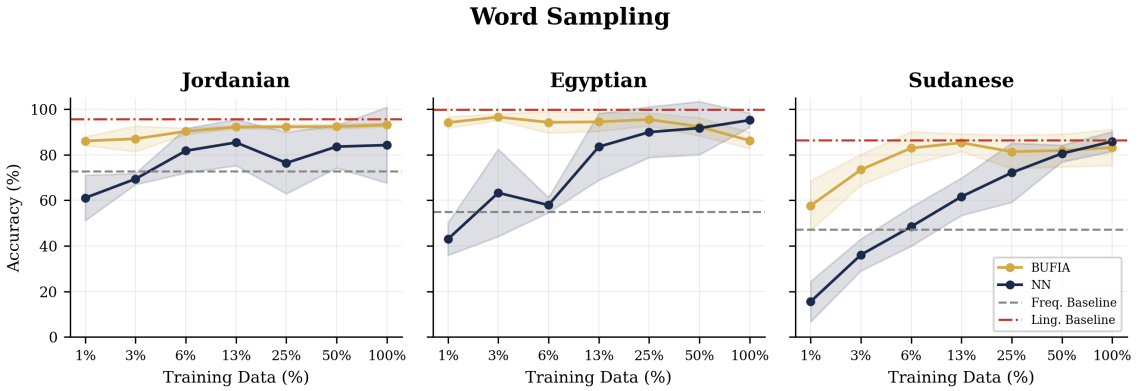


Figure 1: BUFIA compared to NN in word sampling (Sampling Paradigm 1) experiment in three languages.

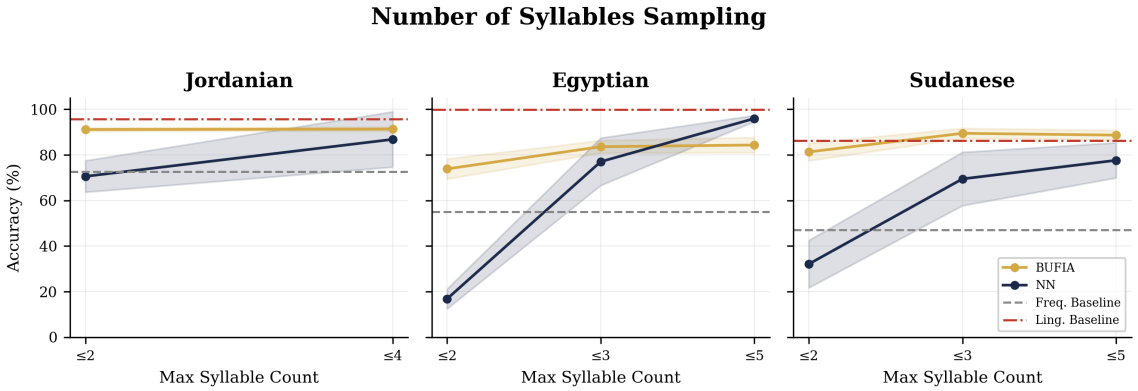


Figure 2: BUFIA compared to NN in the number of syllables sampling experiment (Sampling Paradigm 2) for three languages. The reason why JOR is missing 3 in the graph is because there are only 15 words with 4 syllables in all of the dataset, so we decided to combine them in one step.

The most frequent stress index in Sudanese Arabic is the first syllable with 47.1%. Neither the frequency baseline nor the linguistic rules are trained on our data; therefore, they have no learning curve.

7 Results

The rules derived from the linguistic literature (Section 6.3) achieve very high performance on all three dialects: 99.8% for Egyptian Arabic, in 95.7% for Jordanian Arabic, and 86.3% for Sudanese Arabic.It is worth noting that the linguistic baseline outperformed both models under most conditions. This fact shows that the Arabic stress systems are rule-governed and well studied linguistically.

We can see from Figures 1, 2, and 3 that across all three sampling strategies, BUFIA performs better than the neural network in most settings, especially when training size is minimal. However, for both BUFIA and the neural network, accuracy generally improves with increased training data, and the performance gap between them shrinks. For Egyptian Arabic, we see in Figures 2 and 3 that the neural model outperforms BUFIA at the largest training sizes.

A few other trends stand out. First, the variance between runs of BUFIA is much lower than for the neural network, indicating a relative robustness to the random nature of the data splits. Second, sampling by structure (Sampling Paradigm 3; Figure 3) proves the most challenging for both models, but especially for the neural network, which only surpasses the frequency baseline on Jordanian and Egyptian, and even then, only on the largest training size. Third, we see that BUFIA's performance begins high, and plateaus or even decreases over the course of the two other sampling strategies. The reason for the decrease is that BUFIA is very sensitive to noise and optionality.

8 Error Analysis

In this section, we discuss the errors the different models make while predicting stress. The errors in both NN and BUFIA at the 100% rate are er-

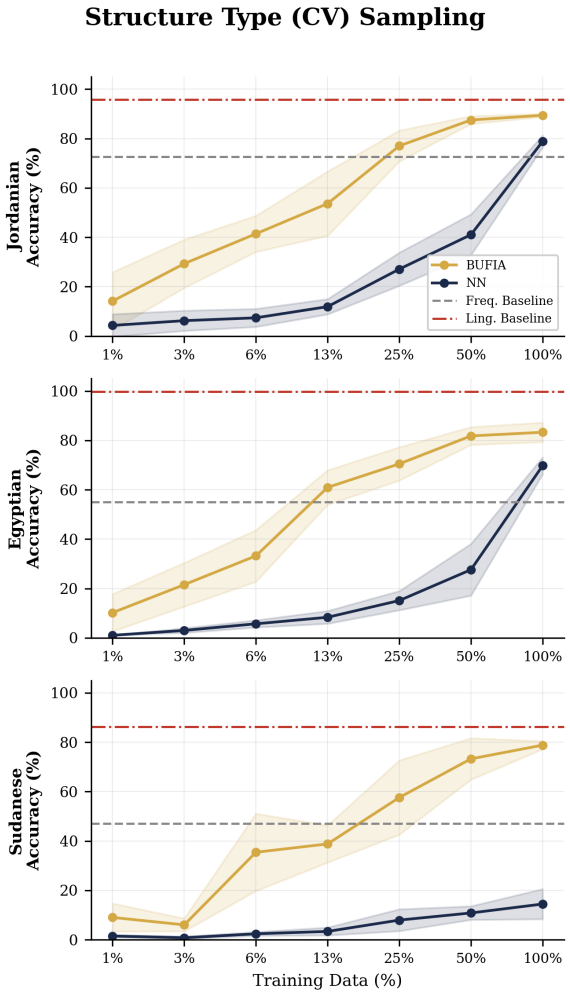


Figure 3: BUFIA compared to NN in structure type (CV) sampling experiment (Sampling Paradigm 3) in three languages.

rors due to exceptions, for example in some cases clitics do not count in the phonemic word but for this setup it will still show as a syllable.Another example, a structure like `cvc.cv.cv` has cases of stress such as `CVC.cv.cv` and `cvc.CV.cv` with the latter being the majority, thus the model gets the `CVC.cv.cv` incorrectly. This is also the case for linguistic rules and it is why the rule-based analysis does not achieve 100% accuracy.

8.1 BUFIA Error Analysis

The errors are due to the training length restriction. BUFIA was trained only on a window of 4 characters due to its exponential big-O time complexity (Chandlee et al., 2019). Thus, it had mistakes such as `CVV.cvvc` instead of `cvv.CVVC` and `CVC.ccvv.cv` instead of `cvc.CCVV.cv`, showing that BUFIA has some issues with 4-segment syllables.

8.2 NN Error Analysis

The types of mistakes that appear in the neural network are changes in syllable-structure identity. For example, outputting `CCCV.cvc` instead of `CCVC.cvc`, where the model swapped `VC` to `CV` in the first syllable. In another case, it outputted `cv.cv.CVC.cv` instead of `cv.cvc.CVV.cv` where a `V` was changed to `C`. There were a few extreme cases of this degenerate performance,for example, in one data point when sampling from structure types, one of the outputs was `cvc.CV.cv.cv.cv

Limitations

We have investigated only three Arabic varieties. The goal of this paper is to pave the way for a broader investigation of a wider range of varieties and to demonstrate that BUFIA can be used for this purpose. We chose the varieties because data were available and because there is scholarly work on these varieties. Extending our work to more varieties is future work. We use the IPA transcription as a starting point, but it is not always available. We will also try to extend the window size for BUFIA to more than four segments.

References

Ali Abdel-Khalig. 2014. *Syllabification and phrasing in three dialects of Sudanese Arabic*. Ph.D. thesis, University of Toronto.

Khaled Hasan AbuAbbas. 2003. *Topics in the phonology of Jordanian Arabic: An optimality theory perspective*. University of Kansas.

Rasheed Saleem Al-Jarrah. 2002. *An optimality-theoretic analysis of stress in the English of native Arabic speakers*. Ball State University.

Khalil Ibrahim Al-Sughayer. 1990. *Aspects of comparative Jordanian and modern standard Arabic phonology*. Michigan State University.

Enam Al-Wer. 2007. Jordanian arabic (amman). *Encyclopedia of Arabic language and linguistics*, 2:505–517.

Abdel-Khalig Ali. 2017. Prosodic domains of syllabification in sudanese arabic. *Perspectives on Arabic Linguistics XXIX*, pages 33–54.

International Phonetic Association. 1999. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press.

Michael Becker. 2022. Cairene arabic stress is local. *Radical: A Journal of Phonology*, 4:211–247.

Ellen Broselow. 2017. Syllable structure in the dialects of arabic. *The Routledge handbook of Arabic linguistics*, pages 32–47.

Ellen Broselow, Su-I Chen, and Marie Huffman. 1997. Syllable weight: convergence of phonology and phonetics. *Phonology*, 14(1):47–82.

Ellen I Broselow. 1976. *The phonology of Egyptian Arabic*. University of Massachusetts Amherst.

Jane Chandlee, Remi Eyraud, Jeffrey Heinz, Adam Jardine, and Jonathan Rawski. 2019. Learning with partially ordered representations. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 91–101, Toronto, Canada. Association for Computational Linguistics.

James Dickins. 2007. *Sudanese Arabic: Phonematics and syllable structure*, volume 38. Otto Harrassowitz Verlag.

Qing Dou, Shane Bergsma, Sittichai Jiampojamarn, and Grzegorz Kondrak. 2009. A ranking approach to stress prediction for letter-to-phoneme conversion. In *Proceedings of the Joint Conference of the 47th annual meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 118–126.

Elan Dresher and Jonathan Kaye. 1990. A computational learning model for metrical phonology. *Cognition*, 34:137–195.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig, editors. 2025. *Ethnologue: Languages of the World*, twenty-eighth edition. SIL International, Dallas, Texas.

Awn al-Sharif Gasim. 1965. Some aspects of sudanese colloquial arabic. *Sudan notes and records*, 46:40–49.

Diana Geneva, Georgi Shopov, Kostadin Garov, Maria Todorova, Stefan Gerdjikov, and Stoyan Mihov. 2023. Accentor: An explicit lexical stress model for tts systems. In *Proc. Interspeech 2023*, pages 4848–4852.

John Goldsmith. 1994. A dynamic computational theory of accent systems. In Jennifer Cole and Charles Kisseberth, editors, *Perspectives in Phonology*, pages 1–28. Stanford: Center for the Study of Language and Information.

John Goldsmith. 2011. The syllable. *The handbook of phonological theory*, pages 164–196.

Matthew K Gordon and Harry van der Hulst. 2020. Word-stress systems.

Prahlad Gupta and David Touretzky. 1991. What a perceptron reveals about metrical phonology. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pages 334–339.

Abdel Halim M Hamid. 1984. *A descriptive analysis of Sudanese colloquial Arabic phonology*. University of Illinois at Urbana-Champaign.

Bruce Hayes. 1995. *Metrical stress theory: Principles and case studies*. University of Chicago Press.

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

Jeffrey Heinz. 2009. On the role of locality in learning stress patterns. *Phonology*, 26(2):303–351.

Bruno Herin, Igor Younes, Enam Al-Wer, and Youssef Al-Sirour. 2021. The classification of bedouin arabic: Insights from northern jordan. *Languages*, 7(1):1.

Saussan Khalil. 2011. Talk like an Egyptian: Egyptian Arabic as an option for teaching communicative spoken Arabic.

Hanaa Kilany, Hassan Gadalla, Howaida Arram, Ashraf Yacoub, Alaa El-Habashi, and Cynthia McLemore. 2002. Egyptian colloquial arabic lexicon. https://catalog.ldc.upenn.edu/LDC99L22. LDC Catalog No. LDC99L22.

Jordan Kodner. 2016. Simple Syllabify.

P Ladefoged and K Johnson. 2014. A course in phonetics, 7th edn. cengage learning. *Inc., Boston*.

Dakotah Lambert. 2024. System description: A theorem-prover for subregular systems: The language toolkit and its interpreter, plebby. In *International Symposium on Functional and Logic Programming*, pages 311–328. Springer.

Seung Suk Lee, Joe Pater, and Brandon Prickett. 2025. Representing and learning stress in a maxent framework. In *Proceedings of the Annual Meetings on Phonology*, volume 1. University of Massachusetts Amherst Libraries.

Bogdan Ludusan, Stefan Ziegler, and Guillaume Gravier. 2012. Integrating stress information in large vocabulary continuous speech recognition. In *INTERSPEECH-Annual Conference of the International Speech Communication Association*, page na.

Stefano Manfredi. 2015. Ethnolinguistic identity and language revitalization among the laggorí in the nuba mountains. *Multidimensional Change in Sudan (1989–2011): Reshaping Livelihoods, Conflicts and Identities*, page 281.

John J McCarthy. 1979. On stress and syllabification. *Linguistic inquiry*, 10(3):443–465.

Abdel Rahman Mustapha. 1982. *La phonologie de l'arabe soudanais (phonématique et accentuation, Tome 1)*. Ph.D. thesis, PhD thesis: Paris: Université de la Sorbonne Nouvelle.

Heikki Palva. 1984. A general classification for the arabic dialects spoken in palestine and transjordan. *Studia Orientalia Electronica*, 55:357–376.

Brandon Prickett and Joe Pater. 2025. Learning and generalizing stress patterns with a sequence-to-sequence neural network. *Linguistics Vanguard*, (0).

Abdelrahim Qaddoumi, Jordan Kodner, Salam Khalifa, Ellen Broselow, and Owen Rambow. 2026. Syllable structures across Arabic varieties. In *Proceedings of the 13th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 250–260, Rabat, Morocco. Association for Computational Linguistics.

Jonathan Rawski. 2021. *Structure and Learning in Natural Language*. Ph.D. thesis, Stony Brook University.

James Rogers and Dakotah Lambert. 2019. Some classes of sets of structures definable without quantifiers. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 63–77, Toronto, Canada. Association for Computational Linguistics.

Jonathan Sakunkoo and Annabella Sakunkoo. 2025. Lost and found: Computational quality assurance of crowdsourced knowledge on morphological defectivity in wiktionary. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 998–1003.

Paul Schachter and Fe T Otanes. 1983. *Tagalog reference grammar*. Univ of California Press.

Kari Suomi, Juhani Toivanen, and Riikka Ylitalo. 2009. *Finnish sound structure: Phonetics, phonology, phonotactics and prosody*. University of Oulu.

Logan Swanson, Jeffrey Heinz, and Jon Rawski. 2026. Phonotactic learning with structure, not statistics. *Linguistic Inquiry*. In press.

Bruce Tesar and Paul Smolensky. 2000. *Learnability in Optimalty Theory*. MIT Press, Cambridge, MA.

Daan van Esch, Mason Chua, and Kanishka Rao. 2016. Predicting pronunciations with syllabification and stress with recurrent neural networks. In *INTERSPEECH*, pages 2841–2845.

Inga Vendelin. 2010. Perception of predictable stress: A cross-linguistic investigation. *Journal of Phonetics*.

Kees Versteegh. 2014. *Arabic language*. Edinburgh University Press.

Chao Wang and Stephanie Seneff. Lexical stress modeling for improved speech recognition of spontaneous telephone speech in the jupiter domain.

Janet CE Watson. 2002. *The phonology and morphology of Arabic*. Oxford University Press, USA.

Janet CE Watson. 2011. Word stress in arabic.

World Bank. 2024. Population, total — jordan. Accessed 2025-10-05.

Shijie Wu, Ryan Cotterell, and Mans Hulden. 2021. Applying the transformer to character-level transduction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1901–1907.

Yuxiang Zou, Shichao Liu, Xiang Yin, Haopeng Lin, Chunfeng Wang, Haoyu Zhang, and Zejun Ma. 2021. Fine-grained prosody modeling in neural speech synthesis using tobi representation. In *Interspeech*, pages 3146–3150.

A Appendix

LDC symbol	Phonetic description	IPA symbol
C	voiceless glottal stop	ʔ
b	voiced bilabial stop	b
t	voiceless dental stop	t
g	voiced velar stop	g
H	voiceless pharyngeal fricative	ħ
x	voiceless velar fricative	x
d	voiced dental stop	d
r	voiced alveolar flap	ɾ
z	voiced alveolar fricative	z
$	voiceless alveopalatal fricative	ʃ
S	voiceless alveolar velarized fricative	sˤ
D	voiced dental velarized stop	dˤ
T	voiceless dental velarized stop	tˤ
Z	voiced velarized interdental fricative	ðˤ
c	voiced pharyngeal fricative	ʕ
G	voiced uvular fricative	ɣ
f	voiceless labio-dental fricative	f
q	voiceless glottal stop	ʔ
Q	voiceless pharyngeal stop	—
k	voiceless velar stop	k
l	voiced alveolar lateral	l
m	voiced bilabial nasal	m
n	voiced alveolar nasal	n
h	voiceless glottal fricative	h
w	voiced bilabial continuant	w
y	voiced palatal continuant	j
v	voiced labio-dental fricative	v
j	voiced alveopalatal affricate	dʒ

Table 3: Phonology table of the LDC Arabic lexicon and its corresponding IPA — consonants and glides

LDC symbol	Phonetic description	IPA symbol
@	low front unrounded vowel	ɑ
a	low back unrounded vowel	a
i	high front unrounded vowel	i
u	high back rounded vowel	u
%	long @	ɑɑ / ɑː
A	long a	aa / aː
I	long i	ii / iː
O	long back mid rounded vowel	oo / oː
U	long u	uu / uː
E	long front mid unrounded vowel	ee / eː
ay	front upgliding diphthong	aj
aw	back upgliding diphthong	aw

Table 4: Vowels and diphthongs

	c	**v**	**C**	**V**	**#**	**0**
stress	−	−	+	+	0	0
consonant	+	−	+	−	0	0
vowel	−	+	−	+	0	0
wb	0	0	0	0	+	0
sb	0	0	0	0	0	+

Table 5: BUFIA feature matrix: c, s = unstressed consonant/vowel; C, V = stressed counterparts; wb = word boundary; sb = syllable boundary. Symbols: $+$ present, $-$ absent, 0 not applicable.

#	Factor 1	Factor 2	Factor 3	Factor 4
1	[+sb]	—	—	—
2	[-stress]	[+stress,+vowel]	—	—
3	[+stress]	[-stress,+vowel]	—	—
4	[+wb]	[+vowel]	—	—
5	[+wb]	[+wb]	—	—
6	[ ]	[+wb]	[ ]	—
7	[-stress]	[+stress]	[-stress]	—
8	[-stress]	[+stress]	[+consonant]	—
9	[-stress]	[+stress]	[+wb]	—
10	[+stress]	[-stress]	[+stress]	—
11	[+stress]	[-stress]	[+consonant]	—
12	[+stress]	[-stress]	[+wb]	—
13	[+consonant]	[+consonant]	[+stress,+consonant]	—
14	[+consonant]	[-stress,+consonant]	[+consonant]	—
15	[+consonant]	[-stress,+consonant]	[+wb]	—
16	[+vowel]	[+vowel]	[+vowel]	—
17	[+vowel]	[-stress,+vowel]	[+stress]	—
18	[+vowel]	[-stress,+vowel]	[+wb]	—
19	[+wb]	[-stress]	[+stress]	—
20	[+wb]	[+stress]	[-stress]	—
21	[+stress,+vowel]	[ ]	[+stress,+vowel]	—
22	[-stress]	[ ]	[+stress]	[+wb]
23	[+stress]	[ ]	[-stress]	[+stress]
24	[+stress]	[ ]	[+consonant]	[+stress,+vowel]
25	[+stress]	[+consonant]	[+stress,+vowel]	[-stress]
26	[+stress]	[+consonant]	[+stress,+vowel]	[+wb]
27	[+vowel]	[ ]	[+consonant]	[+stress,+consonant]
28	[+vowel]	[ ]	[-stress,+consonant]	[+consonant]
29	[+vowel]	[ ]	[+stress,+vowel]	[-stress]
30	[+wb]	[ ]	[-stress]	[+wb]
31	[+wb]	[ ]	[+consonant]	[+consonant]
32	[+wb]	[ ]	[+consonant]	[+wb]
33	[+wb]	[ ]	[+vowel]	[-stress,+vowel]
34	[-stress,+consonant]	[ ]	[+vowel]	[+wb]
35	[-stress,+consonant]	[ ]	[+vowel]	[-stress,+vowel]

Table 6: BUFIA-learned phonotactic constraints with k=4 and n=3 for Jordanian stress. `wb` = word boundary. `sb` = syllable boundary. Empty `[ ]` indicates that any segment is allowed.

Hyperparameter	Value
Batch size	400
Max steps	20,000
Optimizer	Adam (β_1=0.9, β_2=0.98)
Learning rate	0.001 (min LR 1×10^{-5})
Momentum	0.9
Error stop	10^{-8}
LR scheduler	Warmup inverse square; warmup steps = 4000
Label smoothing	0.1
Gradient clipping (max_norm)	1.0
Max input length (`max_seq_len`)	128
Max decode length (`max_decode_len`)	128
Dropout	0.3
Embedding Dimension	256
Decode strategy	Greedy
Beam size	5
Best model selection	By accuracy (`bestacc=True`)
Shuffle	True
Other flags	`cleanup_anyway=True`, `saveall=False`

Table 7: Training and decoding hyperparameters

	Syllable	Jordanian	Egyptian	Sudanese
l/L	CV	[sa.ʔa.lu] “they asked”	[ka.tab] “he wrote”	[ka.tab] “he wrote”
h/H	CVV	[saa.ʔil] “questioner”	[kaa.tib] “writer”	[kaa.tib] “writer”
	CVC*	[sam.ne] “ghee”	[mak.tab] “office”	[mak.tab] “office”
	CCVC	[ʃraħ.ha] “explain it fem”	–	–
x/X	CVCC	[sadd] “he blocked”	–	[bank] “bank”
	CVVC	[faat] “he entered”	[ki.taab] “book”	[baab] “door”
	CCVVC	[klaab] “dogs”	–	–
	CCVCC	[mʃadd] “fastener”	–	–
	CVVCC	[dʒaadd] “serious ms. sg.”	–	[koobs] “electric socket”

Table 8: Example of licit syllable structures in Jordanian, Sudanese and Egyptian. We note that both CVCC and CVVC can occur only in the word-final position for Egyptian (McCarthy, 1979). * CVC’s weight is light if it is in word final position because the last C in CVC is extrametrical (Hayes, 1995). Syllables said to not exist in the variety in the literature are marked with a ‘-’.

Step	Seed	Train	Dev	Test
1	21	26	-	-
	42	26	-	-
	82	26	-	-
	99	26	-	-
	101	26	-	-
2	21	53	-	-
	42	53	-	-
	82	53	-	-
	99	53	-	-
	101	53	-	-
3	21	104	-	-
	42	104	-	-
	82	104	-	-
	99	104	-	-
	101	104	-	-
4	21	216	-	-
	42	216	-	-
	82	216	-	-
	99	216	-	-
	101	216	-	-
5	21	416	-	-
	42	416	-	-
	82	416	-	-
	99	416	-	-
	101	416	-	-
6	21	832	-	-
	42	832	-	-
	82	832	-	-
	99	832	-	-
	101	832	-	-
7	21	1664	209	209
	42	1664	209	209
	82	1664	209	209
	99	1664	209	209
	101	1664	209	209

(a) Jordanian

Step	Seed	Train	Dev	Test
1	21	24	-	-
	42	24	-	-
	82	24	-	-
	99	24	-	-
	101	24	-	-
2	21	48	-	-
	42	48	-	-
	82	48	-	-
	99	48	-	-
	101	48	-	-
3	21	95	-	-
	42	95	-	-
	82	95	-	-
	99	95	-	-
	101	95	-	-
4	21	196	-	-
	42	196	-	-
	82	196	-	-
	99	196	-	-
	101	196	-	-
5	21	378	-	-
	42	378	-	-
	82	378	-	-
	99	378	-	-
	101	378	-	-
6	21	757	-	-
	42	757	-	-
	82	757	-	-
	99	757	-	-
	101	757	-	-
7	21	1514	190	190
	42	1514	190	190
	82	1514	190	190
	99	1514	190	190
	101	1514	190	190

(b) Egyptian

Step	Seed	Train	Dev	Test
1	21	9	-	-
	42	9	-	-
	82	9	-	-
	99	9	-	-
	101	9	-	-
2	21	19	-	-
	42	19	-	-
	82	19	-	-
	99	19	-	-
	101	19	-	-
3	21	38	-	-
	42	38	-	-
	82	38	-	-
	99	38	-	-
	101	38	-	-
4	21	78	-	-
	42	78	-	-
	82	78	-	-
	99	78	-	-
	101	78	-	-
5	21	151	-	-
	42	151	-	-
	82	151	-	-
	99	151	-	-
	101	151	-	-
6	21	302	-	-
	42	302	-	-
	82	302	-	-
	99	302	-	-
	101	302	-	-
7	21	604	76	76
	42	604	76	76
	82	604	76	76
	99	604	76	76
	101	604	76	76

(c) Sudanese

Table 9: Word sampling dataset splits for Jordanian, Egyptian, and Sudanese Arabic.

Step	Seed	Train	Types	Dev	Types	Test	Types
1	42	17	3	-	-	-	-
	82	264	1	-	-	-	-
	99	28	2	-	-	-	-
	101	129	1	-	-	-	-
2	21	65	4	-	-	-	-
	82	269	2	-	-	-	-
	99	61	3	-	-	-	-
	101	157	2	-	-	-	-
3	21	147	5	-	-	-	-
	42	154	6	-	-	-	-
	82	270	3	-	-	-	-
	99	131	6	-	-	-	-
	101	158	3	-	-	-	-
4	21	282	8	-	-	-	-
	42	226	9	-	-	-	-
	82	357	6	-	-	-	-
	99	222	8	-	-	-	-
	101	248	6	-	-	-	-
5	21	648	12	-	-	-	-
	42	500	11	-	-	-	-
	82	501	11	-	-	-	-
	99	501	21	-	-	-	-
	101	598	14	-	-	-	-
6	21	853	30	-	-	-	-
	42	847	24	-	-	-	-
	82	1050	23	-	-	-	-
	99	1065	27	-	-	-	-
	101	899	22	-	-	-	-
7	21	1691	44	208	76	208	76
	42	1691	42	208	76	208	76
	82	1691	42	208	76	208	76
	99	1691	44	208	76	208	76
	101	1691	43	208	76	208	76

Table 10: Jordanian – Structure Type (CVS)

Step	Seed	Train	Types	Dev	Types	Test	Types
1	42	26	1	-	-	-	-
	82	31	1	-	-	-	-
	101	55	1	-	-	-	-
2	21	132	3	-	-	-	-
	82	48	2	-	-	-	-
	99	106	3	-	-	-	-
	101	209	2	-	-	-	-
3	21	179	4	-	-	-	-
	42	143	5	-	-	-	-
	82	140	4	-	-	-	-
	99	142	4	-	-	-	-
	101	275	4	-	-	-	-
4	21	344	8	-	-	-	-
	42	202	8	-	-	-	-
	82	225	8	-	-	-	-
	99	217	9	-	-	-	-
	101	409	8	-	-	-	-
5	21	431	14	-	-	-	-
	42	408	15	-	-	-	-
	82	456	14	-	-	-	-
	99	397	15	-	-	-	-
	101	594	15	-	-	-	-
6	21	783	30	-	-	-	-
	42	826	33	-	-	-	-
	82	778	28	-	-	-	-
	99	817	29	-	-	-	-
	101	851	29	-	-	-	-
7	21	1553	56	189	108	189	108
	42	1547	60	189	108	189	108
	82	1555	54	189	108	189	108
	99	1549	58	189	108	189	108
	101	1549	57	189	108	189	108

Table 11: Egyptian – Structure Type (CVS)

Step	Seed	Train	Types	Dev Types		Test	Types
1	21	23	1	-	-	-	-
	42	29	2	-	-	-	-
	99	134	1	-	-	-	-
	101	19	3	-	-	-	-
2	21	34	2	-	-	-	-
	42	47	4	-	-	-	-
	82	136	2	-	-	-	-
	99	136	2	-	-	-	-
3	21	54	4	-	-	-	-
	82	159	3	-	-	-	-
	99	157	3	-	-	-	-
	101	50	7	-	-	-	-
4	21	102	7	-	-	-	-
	42	132	5	-	-	-	-
	82	248	5	-	-	-	-
	99	162	5	-	-	-	-
	101	101	11	-	-	-	-
5	21	155	10	-	-	-	-
	42	170	9	-	-	-	-
	82	297	9	-	-	-	-
	99	217	9	-	-	-	-
	101	165	16	-	-	-	-
6	21	375	25	-	-	-	-
	42	436	24	-	-	-	-
	82	408	18	-	-	-	-
	99	386	18	-	-	-	-
	101	339	23	-	-	-	-
7	21	617	36	83	83	83	83
	42	617	36	83	83	83	83
	82	617	36	83	83	83	83
	99	617	36	83	83	83	83
	101	617	36	83	83	83	83

Table 12: Sudanese – Structure Type (CVS)

Step	Seed	Train	Types	Dev	Types	Test	Types
1	21	260	1	-	-	-	-
2	21	1348	2	-	-	-	-
	42	1350	2	-	-	-	-
	82	1358	2	-	-	-	-
	99	1362	2	-	-	-	-
	101	1358	2	-	-	-	-
3	21	1666	4	208	4	208	4
	42	1666	4	208	4	208	4
	82	1666	4	208	4	208	4
	99	1666	4	208	4	208	4
	101	1666	4	208	4	208	4

Table 13: Jordanian – Number of Syllables

Step	Seed	Train	Types	Dev	Types	Test	Types
1	21	575	2	-	-	-	-
	42	613	2	-	-	-	-
	82	605	2	-	-	-	-
	99	611	2	-	-	-	-
	101	627	2	-	-	-	-
2	21	1325	3	-	-	-	-
	42	1331	3	-	-	-	-
	82	1334	3	-	-	-	-
	99	1345	3	-	-	-	-
	101	1327	3	-	-	-	-
3	21	1517	5	189	7	189	7
	42	1517	5	189	7	189	7
	82	1516	5	189	7	189	7
	99	1517	5	189	7	189	7
	101	1516	5	189	7	189	7

Table 14: Egyptian – Number of Syllables

Step	Seed	Train	Types	Dev	Types	Test	Types
1	21	310	2	-	-	-	-
	42	299	2	-	-	-	-
	82	296	2	-	-	-	-
	99	310	2	-	-	-	-
	101	308	2	-	-	-	-
2	21	543	3	-	-	-	-
	42	542	3	-	-	-	-
	82	538	3	-	-	-	-
	99	547	3	-	-	-	-
	101	544	3	-	-	-	-
3	21	604	5	76	6	76	6
	42	604	5	76	6	76	6
	82	604	5	76	6	76	6
	99	604	5	76	6	76	6
	101	604	5	76	6	76	6

Table 15: Sudanese – Number of Syllables

		Jordanian (JOR)		**Egyptian (EGY)**		**Sudanese (SUD)**	
Seed	**% Data**	**NN**	**BUFIA**	**NN**	**BUFIA**	**NN**	**BUFIA**
21	1%	44.02	85.65	33.16	94.21	7.89	47.37
21	3%	67.46	77.03	77.89	97.89	34.21	73.68
21	6%	89.00	89.00	59.47	96.84	36.84	71.05
21	13%	93.30	92.34	93.68	97.89	61.84	82.89
21	25%	50.72	91.39	89.47	97.89	77.63	76.32
21	50%	91.87	91.39	68.42	94.21	81.58	81.58
21	100%	90.91	92.82	94.21	79.47	89.47	86.84
42	1%	67.46	83.73	39.47	90.00	18.42	57.89
42	3%	71.29	88.52	62.11	97.37	27.63	67.11
42	6%	83.73	88.52	62.11	98.42	47.37	84.21
42	13%	86.60	91.39	86.32	98.42	55.26	88.16
42	25%	75.60	90.91	97.37	98.42	73.68	88.16
42	50%	71.29	92.34	96.84	96.32	82.89	84.21
42	100%	94.74	93.78	89.47	88.42	86.84	88.16
82	1%	66.03	87.56	42.11	97.37	2.63	68.42
82	3%	73.21	86.60	91.05	96.84	42.11	67.11
82	6%	90.43	91.39	55.79	96.84	42.11	78.95
82	13%	91.87	91.87	95.79	86.84	50.00	78.95
82	25%	88.04	91.87	97.37	90.53	47.37	82.89
82	50%	88.04	91.39	98.42	91.05	80.26	81.58
82	100%	89.47	92.34	97.37	89.47	78.95	78.95
99	1%	71.77	84.21	45.26	95.79	26.32	43.42
99	3%	67.94	88.52	42.63	96.84	30.26	73.68
99	6%	63.16	92.34	52.63	85.26	55.26	89.47
99	13%	65.55	93.30	86.32	94.74	72.37	89.47
99	25%	81.82	93.78	96.84	95.79	85.53	69.74
99	50%	73.21	94.26	97.89	85.26	84.21	69.74
99	100%	51.20	94.26	98.42	85.79	82.89	69.74
101	1%	55.98	89.00	54.74	93.16	22.37	71.05
101	3%	66.99	94.26	42.63	93.68	46.05	85.53
101	6%	82.30	89.95	59.47	93.68	60.53	90.79
101	13%	89.47	91.87	55.26	94.21	68.42	86.84
101	25%	85.65	93.30	68.42	94.74	76.32	89.47
101	50%	93.30	92.34	96.84	94.74	73.68	92.11
101	100%	94.74	92.34	96.32	87.89	90.79	92.11

Table 16: Full results for the Word Sampling experiment. Accuracy (%) for NN and BUFIA across 5 seeds and 3 languages. Columns are grouped by language: Jordanian (JOR), Egyptian (EGY), and Sudanese (SUD).

		Jordanian (JOR)		Egyptian (EGY)		Sudanese (SUD)	
Seed	**% Data**	**NN**	**BUFIA**	**NN**	**BUFIA**	**NN**	**BUFIA**
21	1%	–	–	–	–	1.20	7.23
21	3%	4.81	28.85	3.17	34.92	0.00	2.41
21	6%	9.62	38.94	4.76	48.15	1.20	55.42
21	13%	12.98	60.10	10.58	64.55	3.61	49.40
21	25%	25.00	71.63	21.69	67.72	14.46	73.49
21	50%	28.85	88.94	43.92	85.71	14.46	77.11
21	100%	82.21	89.42	64.02	84.66	9.64	75.90
42	1%	0.96	10.10	1.06	19.58	1.20	10.84
42	3%	–	–	–	–	–	–
42	6%	5.29	40.38	4.76	36.51	2.41	39.76
42	13%	8.17	76.44	6.88	61.38	1.20	39.76
42	25%	20.19	76.92	14.29	62.96	2.41	51.81
42	50%	38.46	88.94	35.98	81.48	6.02	73.49
42	100%	78.37	88.46	69.31	85.71	12.05	79.52
82	1%	12.02	33.65	1.06	10.05	–	–
82	3%	12.98	31.25	1.59	10.58	1.20	8.43
82	6%	13.46	31.25	4.23	26.46	2.41	9.64
82	13%	17.31	43.75	5.29	47.62	6.02	36.14
82	25%	22.60	68.75	11.64	72.49	9.64	72.29
82	50%	37.98	87.50	19.58	80.95	10.84	74.70
82	100%	75.00	90.38	70.90	77.25	12.05	80.72
99	1%	0.96	2.40	–	–	1.20	1.20
99	3%	1.92	14.90	3.17	17.99	1.20	7.23
99	6%	3.37	42.79	6.88	17.46	2.41	27.71
99	13%	10.58	42.79	6.88	63.49	2.41	26.51
99	25%	39.42	85.10	11.11	66.67	3.61	32.53
99	50%	47.12	84.62	16.93	75.66	12.05	57.83
99	100%	80.29	89.90	74.60	80.42	26.51	78.31
101	1%	3.37	10.58	1.06	1.06	2.41	16.87
101	3%	5.29	42.31	4.23	22.75	–	–
101	6%	5.29	53.85	7.94	37.57	3.61	44.58
101	13%	10.58	45.19	12.17	67.72	3.61	42.17
101	25%	28.37	82.69	16.93	82.54	9.64	57.83
101	50%	52.88	87.50	21.69	85.19	10.84	83.13
101	100%	78.37	88.94	69.84	88.36	12.05	79.52

Table 17: Full results for the Structure Type (CV) Sampling experiment. Accuracy (%) for NN and BUFIA across 5 seeds and 3 languages. Columns are grouped by language: Jordanian (JOR), Egyptian (EGY), and Sudanese (SUD). Missing cells indicate that the split was not available.

Language	Seed	Max Syll.	NN Acc	BUFIA Acc
Jordanian (JOR)	21	≤2	70.19	89.90
	21	≤4	95.19	90.38
	42	≤2	77.40	91.35
	42	≤4	95.19	91.35
	82	≤2	75.48	90.87
	82	≤4	88.94	92.31
	99	≤2	57.69	91.35
	99	≤4	62.98	92.31
	101	≤2	72.12	92.31
	101	≤4	91.83	90.38
Egyptian (EGY)	21	≤2	13.23	70.37
	21	≤3	67.72	85.19
	21	≤5	96.30	83.07
	42	≤2	12.70	71.96
	42	≤3	85.19	79.37
	42	≤5	94.18	79.37
	82	≤2	14.81	73.02
	82	≤3	86.24	84.13
	82	≤5	97.88	83.60
	99	≤2	19.05	71.43
	99	≤3	61.38	82.01
	99	≤5	94.71	86.24
	101	≤2	24.34	82.54
	101	≤3	84.66	87.30
	101	≤5	96.83	89.42
Sudanese (SUD)	21	≤2	18.42	76.32
	21	≤3	75.00	86.84
	21	≤5	84.21	85.53
	42	≤2	28.95	86.84
	42	≤3	46.05	90.79
	42	≤5	78.95	92.11
	82	≤2	42.11	77.63
	82	≤3	75.00	86.84
	82	≤5	84.21	88.16
	99	≤2	25.00	82.89
	99	≤3	75.00	90.79
	99	≤5	77.63	89.47
	101	≤2	46.05	82.89
	101	≤3	76.32	92.11
	101	≤5	63.16	88.16

Table 18: Full results for the Number of Syllables Sampling experiment. Accuracy (%) for NN and BUFIA across 5 seeds, 3 languages, and data splits by maximum syllable count.

Rating-based property axes: Agentivity and split intransitivity

Eva Neu
UMass Amherst
eneu@umass.edu

Brian Dillon
UMass Amherst
bwdillon@umass.edu

Katrin Erk
UMass Amherst
kerk@umass.edu

Abstract

Many gradable properties have been found to be encoded as axes in embedding space. Most commonly, property axes are computed using seed words, but recent work has noted limitations to seed-based axes. Here, we present a novel methodology for computing property axes that is based on human ratings and does not require seeds. We apply this methodology to a particular problem at the syntax-semantics interface, namely, the question which semantic properties of intransitive verbs affect their likelihood to occur in one of two syntactic structures, unergative and unaccusative. Comparing property axes that encode different semantic dimensions of the concept of agentivity, we find that properties like movement and being alive are a better predictor of the syntactic behavior of intransitives than goal-directedness or intentionality. We discuss the potential of rating-based axes for future work in semantics and at the syntax-semantics interface.

1 Introduction

1.1 Properties and property axes

In linguistics and many other areas of research, it is often necessary to obtain a measure of a gradable property for a set of words. For instance, one might want to know how dangerous different kinds of animals are considered to be, what degree of wealth is associated with different kinds of sports, or how attractive people find different cities. Traditionally, such measures have been obtained by collecting ratings for these properties from human participants.

Recent work on word embeddings has shown that gradable properties are also encoded by *property axes* in embedding space (e.g., Grand et al. 2022; Kozlowski et al. 2019; Garí Soler and Apidianaki 2020; Lucy et al. 2022). The projection of a word embedding on such an axis encodes to what extent the word is associated with the positive or the negative end of this semantic dimension. The most widely used method for obtaining axes for gradable properties is through manually defined seed words. For example, for *danger*, the two poles of the property could be described with {*dangerous*, *unsafe*} and {*safe*, *harmless*}, respectively. An axis for *danger* can then be obtained as the mean over all difference vectors $\vec{safe} - \vec{dangerous}$, $\vec{safe} - \vec{unsafe}$, $\vec{harmless} - \vec{dangerous}$, $\vec{harmless} - \vec{unsafe}$. How close the projection of a word embedding sits towards the *dangerous*/*unsafe* end of the axis then serves as a measure of how dangerous the entity denoted by the word can be considered to be.

However, while simple and easy to apply, seed-based property axes face problems (Erk and Apidianaki, 2024; Lucy et al., 2022; Antoniak and Mimno, 2021). Their performance in capturing human judgments depends critically on which seeds are chosen. However, there is no agreed-upon method for finding seed words, and in practice, researchers adopt a wide range of strategies for settling on a set of seeds (e.g., selecting them by hand, crowdsourcing, etc.). The repercussions of these different methodological choices are unclear. It is rare for different sets of seed words to be compared to each other in order to determine which seeds are more or less appropriate for the property in question. Furthermore, the performance of seeds can be affected by word frequencies (Ethayarajh et al., 2019), and they can reflect the researchers' underlying social and cultural biases.

Against this background, this study develops a new methodology for computing property axes. These axes are based on human judgments but can be used to compute measures for words for which no human ratings have been obtained. We apply this methodology to a particular problem in linguistic theory, the semantic correlates of split intransitivity, which we introduce next.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 280–290
July 3-4, 2026 ©2026 Association for Computational Linguistics

1.2 Split intransitivity

It has long been known that the syntactic structures in which a verb can appear depend on its lexical semantics (e.g., Dowty 1991; Jackendoff 1983; Levin 1993; Levin and Rappaport Hovav 1995; Van Valin Jr 1990). For instance, verbs denoting a transfer of possession such as *give* typically appear in a ditransitive syntax, while verbs denoting a causal relation such as *kill* tend to surface in a transitive syntax.

One area in which the effect of lexical semantics on syntax has been discussed most extensively is split intransitivity. Intransitive verbs are commonly assumed to allow for two different underlying syntactic structures, unergative and unaccusative, which can be distinguish from each other based on a wide variety of syntactic diagnostics (Burzio, 1981, 1986; Perlmutter, 1978). For instance, *freeze* but not *play* can surface in prenominal participle constructions (1):

(1) a. the frozen lake
b. *the played child

Moreover, *freeze* but not *play* can take a secondary predicate describing the result of the event (2):

(2) a. The lake froze solid.
b. *The child played tired. (i.e., became tired as the result of playing)

These contrasts can be captured by positing two different syntactic structures for intransitives, unergative and unaccusative. Unaccusative structures allow for prenominal participle constructions and resultative predicates; unergative structures do not. In (1) and (2), *freeze* behaves as an unaccusative and *play* as an unergative.

The unergative/unaccusative split is not a neat divide between two distinct classes of verbs. Rather, verbs have a gradient preference towards one structure or the other, with some leaning strongly towards an unergative structure, others towards an unaccusative structure, and others being more variable in their syntactic behavior (Sorace, 2000, 2011, 2004). Crucially, this tendency in a verb's syntactic behavior has been linked to its semantics: verbs that denote a more *agentive* activity appear to be more likely to be used as unergatives. For instance, *play* describes a more agentive activity than (intransitive) *freeze*, thus favouring an unergative structure.

Theoretically, such a correlation between an agentive meaning and an unergative syntax has been motivated based on *thematic roles*. The Unaccusativity Hypothesis developed in the generative tradition holds that in an unergative syntax, the sole argument is merged in the same position as the subject argument of transitives, whereas in an unaccusative syntax, it starts out in the same position as the object argument of transitives (Burzio 1981, 1986; Perlmutter 1978, but see Van Valin Jr 1990). This analysis suggests that the arguments of unergatives and unaccusatives should receive the same thematic role as the subject and object argument of transitives, respectively, with the argument of unergatives being assigned the thematic role of an Agent, and the argument of unaccusatives the thematic role of a Patient. Thus, a verb should be the more likely to adopt an unergative syntax the more agentive its meaning is.

Empirically, however, the link between agentivity and unergativity remains contested. It has been documented most extensively in a series of papers by Sorace (2000, 2011, 2004), who grouped verbs into seven different classes based on their degree of agentivity – besides another semantic property, telicity – and argued that more agentive verbs are more likely to behave as unergatives (see also Acartürk and Zeyrek 2010; Allman 2017; Baker 2019; Huang 2018 for similar, smaller-scale studies). However, a major limitation to Sorace's work is that her classification of verbs into semantic groups was only based on intuition. Testing if – and if so, how strongly – agentivity predicts the syntactic behavior of intransitives requires an actual quantitative measure of agentive semantics rooted in empirical data.

In recent work, Kim et al. (2024) aimed to fill this gap by conducting a systematic comparison of various semantic measures to determine their predictive value for the syntactic behavior of English intransitives. These measures include Sorace's seven verb classes as well as a similar verb classification developed by Levin (1993). Another set of predictors was derived from the GloVe embeddings of the verbs by means of a Principal Component Analysis. In addition, Kim et al. collected semantic ratings for each verb on a scale from 0 to 6 for two sets of features. One consisted of 6 event-related features traditionally considered relevant for the unergative/unaccusative distinction (Agentivity, Telicity, Caused, Transitivity, Dynam-

icity, Requires energy input). The other contained a set of 66 properties developed by Binder et al. (2016) such as Color, Pain, Duration and Angry. Binder and colleagues argue that these 66 experiential features each have a distinct neurological representation and can thus jointly capture how concepts are represented in the brain.

To evaluate the performance of these predictors, Kim et al. collected ratings for 138 verbs on how acceptable they are in reduced relative clauses, an unaccusativity diagnostic (see (1)). Ratings are expected to be higher for more strongly unaccusative verbs. Each verb was presented in the context of three different phrases and rated on a 1–5 Likert scale. The event-related features predicted the syntactic data better than Levin's and Sorace's categorizations, but only Caused emerged as a significant predictor after correcting for multiple comparisons. The model combining experiential and event-related features predicted the syntactic ratings best; Caused and Agentivity were significant predictors among several others. The GloVe-based model came in second. Kim et al. concluded that the unergative/unaccusative distinction is best described as being rooted in graded, embodied features of sensory experience, as formalized in the set of properties developed by Binder et al. (2016). This picture suggests that the effect of agentivity on the syntax of intransitives might have been overestimated.

However, we see a caveat to this finding. Agentivity is a broad and multifaceted concept, having been associated with a wide range of properties like intentionality, animacy, causal power, volition, sentience, and others (Dowty, 1991). The question that Kim et al. asked speakers to determine the degree of agentivity of a verb was: *To what extent does this verb describe something that is actively or intentionally done?* However, this question might have operationalized agentivity too narrowly. In other words, the question is not only whether agentivity matters for the syntactic behavior of intransitives, but also *what kind* of agentivity matters. Moreover, we note that rating tasks are not as straightforwardly applied to verbs as they are to nouns, as it can be difficult to formulate rating questions that target the semantic properties of verbs in a way that is natural for human participants. Here, we aim to address both of these concerns.

1.3 This study

In this study, we develop a novel methodology to set up property axes based on human ratings. We compute axes in embedding space that optimize the fit to human ratings and then project other word embeddings onto them, thus simulating ratings for unseen items. Crucially, our axes achieve a good fit to human ratings even without seed words.

We apply this methodology to the problem of split intransitivity by using a set of human ratings from VanArsdall and Blunt (2022). They collected ratings for 1,200 concrete nouns on eight different dimensions of animacy such as similarity to a person, ability to reproduce and goal-directedness. For each dimension, we computed a property axis that optimizes the fit to VanArsdall and Blunt's noun ratings and projected intransitive verbs onto it to derive a semantic measure of the verbs on this property. Rating-based property axes thus allow us to explore different ways of conceptualizing agentivity, while also circumventing the methodological challenges for collecting human ratings for verbs.

To evaluate the performance of these property axes, we tested how well the semantic measures derived from them predict the syntactic measure of unergativity/unaccusativity used by Kim et al. (2024). We find that a broad concept of animacy, being associated with properties like movement and being alive, is a stronger predictor of the unergative/unaccusativity split than a narrow concept of goal-directedness. This also accounts for the fact that Kim et al.'s measure of agentivity, which they operationalized in terms of intentionality, did not perform particularly strongly in their study in predicting the syntactic data.

Our ambition in the following is not to solve the general question of which semantic properties affect the syntactic behavior of intransitives. Rather, the more modest goal of this study is to focus on one semantic property – agentivity – that has emerged from the previous literature as the most likely contender to affect to syntactic behavior of intransitives and determine what kind of agentivity is the strongest predictor for the unergative/unaccusative distinction. We hope that our findings stimulate further research on the semantic correlates of split intransitivity, comparing agentivity to a wider range of semantic properties.

The major contribution of this study, however, is methodological. We add to a growing literature on automatically predicting feature norms (see,

e.g., Baroni and Lenci 2010; Johns and Jones 2012; Herbelot and Vecchi 2015; Făgărăşan et al. 2015; Gupta et al. 2015; Rubinstein et al. 2015; Wang et al. 2017; Chersoni et al. 2021; Rosenfeld and Erk 2023; Utsumi 2020; Chronis et al. 2023). We propose a new technique for extrapolating semantic norms using axes in embedding space which we believe holds a lot of promise for future applications. Overall, our rating-based property axes constitute a low-resource strategy for high-quality simulation of human ratings, making them a fruitful method for a wide range of research projects.

2 Modeling

2.1 Methods

Many gradable properties seem to be linearly encoded in embedding space, a fact that has been used for analyses in linguistics, cognition, and social sciences (Bolukbasi et al., 2016; Kozlowski et al., 2019; Garí Soler and Apidianaki, 2020; Grand et al., 2022). It has even been proposed that many high-level concepts are encoded linearly (Park et al., 2024), although this does not hold for all concepts (Engels et al., 2025). Traditionally, axes in embedding space that encode gradable properties have been computed using seed words (e.g., Grand et al. 2022; Kozlowski et al. 2019; Garí Soler and Apidianaki 2020; Lucy et al. 2022). However, Erk and Apidianaki (2024); Lucy et al. (2022); Antoniak and Mimno (2021) highlight the limitations of this method, as discussed earlier in Section 1.1.

To improve the accuracy of seed-based axes, Erk and Apidianaki (2024) (below: EA) introduced a method for computing property axes that interpolates seed words with training data in the form of human ratings. To compute an axis $\vec{f}$ for a property f of concepts w, they used a loss function that penalizes the predicted scalar projection of $\vec{w}$ onto $\vec{f}$ for deviation from the gold rating y_w for w. This was combined with a second loss that enforces closeness to a seed-based axis.

The EA model optimizes pointwise fit to human ratings. But the typical use of property axes is to predict *rankings* of concepts along the property. We introduce a new model that, like the EA model, fits a property axis to human ratings, but uses a ranking loss to more directly approximate the characteristic of interest and as a result does not require seeds to fit the data well. We use a margin ranking loss (Nayyeri et al., 2019): for any pair (a, b) of concepts where the gold rating of b is higher than that of a, it encourages the predicted value for b to be higher than a's by at least a margin of d:

$$J_r = \sum_{(a,b)\in P, \hat{y}_b > \hat{y}_a|} max(0, d - y_b + y_a)$$

where P is a set of training item pairs, an N-size set sampled from all training item pairs with at least a difference of d in their gold ratings; for a concept pair (a, b), $\hat{y}_a, \hat{y}_b$ are gold ratings, and y_a, y_b are predictions. N and d are the parameters of the model.

2.2 Evaluation

We evaluated the ranking loss model on the large collection of concepts and human property ratings introduced by Grand et al. (2022) and compared it to the original seed based axes of Grand et al. (2022) and the EA model. Grand et al. (2022) measured performance through correlation (pearson *r*) as well as *pairwise order consistency* (POC), the percentage of test pairs ordered correctly by the model. However, we cannot use correlation because the data sets become too small once part of the data is used for training. We measured POC as well as *extended* POC (XPOC), the percentage of test pairs and train/test pairs ordered correctly by the model (Erk and Apidianaki, 2024) – this checks whether test items are ranked correctly with respect to training or test items. Results from a 5-fold crossvalidation on the Grand et al. data are shown in Table 1. *Ranking* is our new model. In the evaluation, as throughout in this paper, we used GLoVE embeddings with 300 dimensions pre-trained on Wikipedia and Gigaword.[1]

model	POC	XPOC
seed	.629	.631
pointwise	.701	.779
ranking	.703	.797

Table 1: Comparing our ranking-based fitted axes to EA (pointwise) and seed-based axes on the data of Grand et al. (2022). Pointwise uses interpolated losses based on human ratings and seed axes. All values are averaged over folds and conditions.

We see that the ranking loss model, like the EA model, clearly improves fit over the seed-based

[1]Hyperparameters of the EA model are as given in that paper; hyperparameters for our ranking model were optimized on the development portion of the Grand et al. (2022) data defined by Erk and Apidianaki (2024), $d = 0.2 \cdot$ sdev, $N = 300$.

axes, and achieves even better performance than the EA model. The EA model interpolates human ratings with seeds, and in fact flounders when no seed axis is given. The ranking model, in contrast, manages to fit the data very well even without the help of a seed axis.

To extend our evaluation of the ranking loss model, we evaluated on the VanArsdall and Blunt (2022) dataset which we draw on later to predict the syntactic behavior of intransitives. VanArsdall and Blunt collected ratings for 1,200 concrete nouns on 6 different animacy dimensions: general living/non-living scale, ability to think, ability to reproduce, similarity to a person, goal-directedness and movement likelihood. Using factor analysis, these axes were then clustered into two coarser dimensions, mental and physical animacy. Table 2 shows the highest-scoring nouns for each animacy dimension in their study. Note that while VanArsdall and Blunt refer to the semantic properties measured as variants of *animacy*, they are broad enough to also be describable as *agentivity*. In the following, we largely use the two terms interchangeably.

This dataset differs from Grand et al. (2022) in that first, it contains much more data for each individual property. This allows us to compute the correlation between gold ratings and predictions because the test folds of the cross-validation are larger. Second, the VanArsdall and Blunt dataset does not come with seeds. Hence, we compare the ranking loss model to the EA pointwise loss model without seeds. Hyperparameter values for the EA and ranking loss models were not fit anew; we reused the values fit to the Grand et al. data by EA.

Table 3 summarizes the results of a 5-fold cross-validation on the VanArsdall and Blunt (2022) data. We omit XPOC, which patterns with POC, and instead report correlation. We find that the ranking loss model again achieves a solid fit to the data, while the pointwise loss model performs poorly in the absence of seeds.

feature	POC	pearson *r*
pointwise	.516	0.05
ranking	.783	.789

Table 3: Crossvalidating the fitted axes on the VanArsdall and Blunt (2022) data, comparing pointwise and ranking loss. Values are averaged over folds and conditions. Both sets of axes are fitted without seeds.

Tables 4 and 5 show the performance of the ranking loss model and the pointwise loss model for each animacy dimensions. Again, the ranking loss model is superior: with the exception of Thought, it scores over .7 on pearson *r* for all dimensions, whereas the pointwise loss model only performs well for Thought, Mental and Physical, with values close to or below 0 in the other categories.

feature	POC	pearson *r*
Living	.791	.788
Thought	.505	.024
Repr.	.789	.800
Person	.779	.801
Goals	.796	.800
Move	.764	.740
Mental	.811	.808
Physical	.794	.808

Table 4: Crossvalidating axes fitted with ranking loss (no seeds) on the VanArsdall and Blunt (2022) data, results for individual dimensions. Values are averaged over folds and conditions.

Living	Thought	Reproduction	Person	Goals	Move
waitress	expert	physician	referee	president	twister
waiter	engineer	mother	boyfriend	leader	squirrel
son	human	mom	teenager	governor	waitress
officer	biologist	dog	salesman	physician	ambulance
kitten	astronaut	human	person	doctor	cougar
frog	woman	walrus	officer	inventor	tornado
elephant	surgeon	toad	man	astronaut	bunny
cousin	leader	cheerleader	human	attorney	wasp
citizen	captain	bee	detective	scientist	politician
chipmunk	professor	man	uncle	detective	runner

Table 2: Highest-scoring nouns per animacy dimension in the VanArsdall and Blunt (2022) dataset.

feature	POC	pearson r
Living	.503	.002
Thought	.780	.809
Repr.	.543	.128
Person	.520	.077
Goals	.533	.098
Move	.493	-.016
Mental	.811	.808
Physical	.794	.808

Table 5: Crossvalidating axes fitted with pointwise loss (no seeds) on the VanArsdall and Blunt (2022) data, results for individual dimensions. Values are averaged over folds and conditions.

We further tested the performance of the ranking loss axes by taking the 60,000 most frequent words in COCA, extracting the verbs from that list, and among them, computing the words with the highest values on each property axis for the six fine-grained animacy dimensions. The results are summarized in Table 6. For all axes, the highest-scoring words are indeed associated with animacy/agentivity, but we also see that the axes single out different shades of this concept. For example, *infected* is only in the top 10 verbs for the dimensions Living and Reproduction, being associated with biological life rather than human action more specifically. Similarly, *preys* only scores highly for Reproduction. *Graduate* occurs exclusively in the top 10 verbs for Thought, Person and Goals; *interviewed* for Thought and Person.

Moreover, we computed the pairwise cosine similarity between the eight different animacy axes, summarized in Table 7. We find that dimensions which we would expect to be semantically similar have a higher cosine similarity. For instance, the strongest similarities are between Living and Physical (.967), Reproduction and Physical (.956), and Goals and Mental (.954). In contrast, we find lower similarities between Goals and Reproduction (.708) and Goals and Move (.71): whether an entity acts in a goal-directed fashion and whether it can move and reproduce are clearly very difference concepts. Overall, these findings match how we would expect the different animacy dimensions to compare to each other.

Lastly, we compute the corelation between the eight animacy axes and Kim et al.'s agentivity ratings, for which subjects were asked how intentionally an action was performed. We find a significant correlation for Goals (.286, p-val. .0), Thought (.191, p-val. .001), Person (.191, p-val. .001), and coarse-grained Mental Animacy (.254, p-val. .0), but not for the Physical Animacy features. This is in line with the fact that Kim et al.'s rating task targeted a particular dimension of agentivity, matching most closely VanArsdall and Blunt's category of goal-directedness. Property axes allow us to test a wider spectrum of semantic dimensions with respect to how well they predict the syntactic behavior of intransitives.

3 Predicting the syntactic data

We evaluated how well the eight animacy axes predict the unergative/unaccusative split using the experimental data collected by Kim et al. (2024). Recall that Kim et al. tested the acceptability of 138 intransitive verbs in reduced relative clauses, rated on a 5-point Likert scale. Each verb was shown in the context of three different phrases (e.g., *the frozen ground*, *the frozen lake*, *the frozen popsicles*). Since reduced relatives are an unaccusativity diagnostic, and since we expect animacy/agentivity to be correlated with unergativity, higher animacy scores should correspond to lower acceptability in reduced relatives.

We tested this prediction by fitting mixed-effects Bayesian ordinal regression models with default priors, using the `brms` library in R. All models were computed with a cumulative probit link function and fitted with 2000 iterations (1000 warm-up, 1000 samples taken). R-hat was 1.00 throughout; no divergences were observed during sampling. All gradable features were z-scored. Unlike Kim et al. (2024), we did not average the syntactic ratings over phrases and subjects, which would not allow us to account for inter-subject variation and would generally simplify the data, potentially obscuring important effects. Instead, all our models included by-subject intercepts. The formula for the models is given in (3), with different models differing in the value of animacy_score.

(3) answer $\sim$ animacy_score + (1 | subject)

Table 8 summarizes the fixed effect regression coefficients for each of the eight regression models. As expected, higher animacy scores across all dimensions result in lower acceptability of reduced relative clauses, an unaccusativity diagnostic. All 95% confidence intervals exclude 0.

We then compared the goodness of fit of the dif-

Living	Thought	Reproduction	Person	Goals	Move
infected	acquitted	infected	soldier	killed	killed
emigrated	abducted	preys	acquitted	abducted	soldier
acquitted	killed	emigrated	interviewed	graduate	trained
kill	soldier	kill	graduate	trained	kill
tormented	marshal	acquitted	suspect	soldier	encounters
rat	trained	killed	committed	committed	marshal
killed	interviewed	rat	killed	acquitted	kills
kills	freed	abducted	trained	killing	brave
marshal	graduate	typecast	abducted	confirmed	hates
trained	committed	kills	identified	identified	emigrates

Table 6: Verbs with the highest values on the six fine-grained animacy axes fitted to the VanArsdall and Blunt (2022) dataset with ranking loss. Items are extracted from the list of the 60,000 most frequent words in COCA.

	Living	Thought	Reproduction	Person	Goals	Move	Mental	Physical
Living	1	.846	.949	.806	.712	.801	.783	.967
Thought	.846	1	.849	.924	.927	.803	.957	.85
Reproduction	.949	.849	1	.776	.708	.779	.776	.956
Person	.806	.924	.776	1	.892	.769	.937	.813
Goals	.712	.927	.708	.892	1	.71	.954	.716
Move	.801	.803	.779	.769	.71	1	.786	.819
Mental	.783	.957	.776	.937	.954	.786	1	.795
Physical	.967	.85	.956	.813	.716	.819	.795	1

Table 7: Pairwise cosine similarities between the eight animacy axes fitted to the VanArsdall and Blunt (2022) dataset with ranking loss.

predictor	est.	est. error	l-95%	u-95%
Living	-0.24	0.01	-0.26	-0.21
Thought	-0.23	0.01	-0.25	-0.20
Repr.	-0.20	0.01	-0.22	-0.18
Person	-0.22	0.01	-0.24	-0.20
Goals	-0.14	0.01	-0.16	-0.12
Move	-0.35	0.01	-0.37	-0.32
Mental	-0.20	0.01	-0.23	-0.18
Physical	-0.25	0.01	-0.27	-0.22

Table 8: Estimates for fixed-effect coefficients with estimate error and upper and lower bound 95% confidence interval.

ferent models with a leave-one-out (LOO) analysis using the `loo` library in R. LOO provides a measure of predictive accuracy by training a model on all data points except one and then testing how well the model predicts the held-out data point. Each model is compared to a null model that only includes by-subject intercepts (4):

(4) answer $\sim$ (1 | subject)

This allows us to evaluate how well the two semantic measures predict the syntactic data relative to each other.

Table 9 summarizes the result of the LOO analysis in terms of expected log predictive density (ELPD). ELPD is a measure of the log probability that the model attributes to all the held-out data points; a higher value (closer to zero) corresponds to better predictive performance.

models	ELPD diff	SD
Move vs. null model	444.6	28.5
Living vs. null model	214.4	2.6
Thought vs. null model	195.5	19.2
Person vs. null model	184.3	18.8
Reproduction vs. null model	149.2	17.2
Goals vs. null model	74.3	12.1
Mental vs. null model	156.3	17.4
Physical vs. null model	23.5	21.2

Table 9: LOO analysis with different animacy axes.

For the six fine-grained axes, we see that the strongest improvement over the null model is achieved by Move (444.6), followed by Living (214.4). Goals occupies the very bottom of the

list (74.3). This indicates that a broader notion of animacy associated with properties like movement and being alive is a better predictor for the syntactic behavior of intransitives than a narrow notion of goal-directedness or intentionality.

However, for the coarse-grained axes, Physical Animacy performs better than Mental Animacy. At first, this finding seems to point in the opposite direction than the results from the fine-grained axes. We argue that this apparent contradiction is related to some inherent limitations to the Physical dimension itself. Recall that VanArsdall and Blunt (2022) computed the two coarse-grained animacy axes via factor analysis. While taken together, these two axes explain a total of 86.07% of the variance in the data, 79.05% were accounted for by Mental, and only 7.02% by Physical. Moreover, the factor loading of Move – the strongest predictor among the narrow dimensions – on Physical is fairly low at .56, contrasting with .96 for Living and .93 for Reproduction. This is also reflected in the pairwise cosine similarities between the axes in Table 7. Overall, Physical does not predict much of the variance in the dataset in general, and it is only moderately related to the dimension of Move. We conclude that the results from the fine-grained animacy dimensions are more informative for our purposes.

4 Discussion

This study has developed a novel method for computing property axes in embedding space based on human ratings using a ranking loss function. We have evaluated the ranking loss axes on two large-scale datasets, Grand et al. (2022) and VanArsdall and Blunt (2022), in both cases finding a solid fit to the data. For both datasets, the ranking loss axes clearly outperformed rating-based property axes computed with pointwise loss (Erk and Apidianaki, 2024), which required seed words to match human ratings well.

We have then applied this methodology to the problem of determining the semantic correlates of the unergative/unaccusative contrast. We have focused on the concept of agentivity/animacy as the most likely contender for a semantic property that affects split intransitivity, and investigated which dimension of agentivity/animacy best predicts a measure of the syntactic behavior of intransitive verbs, namely, their acceptability in reduced relative clauses. We found that a lower-level concept of animacy, associated with properties such as movement and being alive, emerged as a stronger predictor than a narrow notion of goal-directedness. The vastly different performance of the different property axes highlights the importance of carefully operationalizing agentivity/animacy by taking into account the various semantic dimensions associated with it, rather than equating it with a single, more specific concept like intentionality.

Our findings for split intransitivity come with a number of limitations. First, in terms of the syntactic data used here, we have tested the performance of our property axes for a single unaccusativity diagnostic, reduced relative clauses. Adding further unaccusativity diagnostics such as resultative secondary predicates or agent nominalizations would give us a more complete picture.

Second, the GloVe embeddings we used represented each verb with a single word form, namely the unmarked form (e.g., *run*). The same form was also used in Kim et al.'s experiment. However, it would be more accurate encode the verb by abstracting over its different morphological forms (e.g., *run*, *runs*, *ran* and *running*). In future work, we aim to represent verbs as sets or clusters of embeddings, each encoding a different morphological form, rather than as a single embedding. This is especially imperative for morphologically richer languages in which verbs can take on dozens of different forms.

Third, we have focused here exclusively on the concept of agentivity/animacy. However, the syntactic behavior of unergatives might be equally, or perhaps even more strongly, affected by other semantic properties, with especially telicity being another promising candidate (e.g., Sorace 2000, 2011, 2004). We hope that by exploring the different dimensions of agentivity, this paper has established a more solid ground for future studies to further compare agentivity to other properties. This might also take the form of exploring the embedding space more directly: rather than identifying an axis for a particular property that is hypothesized to predict the syntactic data and then test its performance, we might also identify an axis that performs well in predicting the syntactic data and then attempt to determine the semantic property associated with it.

Fourth, a confound comes from the fact that many verbs in our sample of intransitives – e.g., *break* – can also be used as transitives, as in *The glass broke/I broke the glass*. For predicting the

syntactic behavior of the intransitive, we are interested in the degree of animacy associated with *the glass*. However, projecting the embedding for *break* onto an animacy axis does not allow us to specifically target properties associated with one argument rather than another, thus potentially distorting the semantic measures derived from it. A related issue is that some English verbs double as nouns: in Table 6, which reported the highest-scoring verbs for each animacy dimensions, several items arguably made it to the top of the list not because of their verbal, but because of their nominal use (e.g., *soldier*, *marshal*, *rat*). In response to this predicament, for the future we plan to extend our work to token-level embeddings, which encode verb meaning in context. This would allow us to specifically extract intransitive verbs and test their behavior with respect to property axes.

Lastly, another direction for future research lies in broadening the scope of our work beyond English. Investigating to what extent split intransitivity is sensitive to the same semantic properties cross-linguistically would also tie in with research on the cognitive status of thematic roles such as Agent and Patient (see Rissman and Majid 2019 for an overview), which the unergative/unaccusative split has been argued to be based on.

Beyond split intransitivity, we believe that rating-based property axes hold much promise for research in semantics and at the syntax-semantics interface. Compared to seed-based axes, they do not require selecting a set of seed words manually, which is prone to various biases and confounds, but allow us to directly recover axes in embedding space that map onto human judgments. Rating-based axes thus constitute a low-resource strategy for high-quality simulation of human ratings. A methodological question to address in the future is how much data in the form of human ratings is needed to compute axes that perform well. Moreover, the axes we have used here were both trained and cross-validated on nouns, but we then extrapolated ratings from them to verbs. Further research should address whether the application of property axes across categories poses any special challenges. Overall, we hope that our work contributes to a better understanding of how semantic properties are encoded in embedding space.

Acknowledgments

We thank Songhee Kim for sharing her data with us.

References

Cengiz Acartürk and Deniz Zeyrek. 2010. Unaccusative/unergative distinction in Turkish: A connectionist approach. In *Proceedings of the 8th Workshop on Asian Language Resources*, pages 111–119, Beijing, China.

JungAe Lee Allman. 2017. *Empirical examination of two diagnostics of Korean unaccusativity*. Ph.D. thesis, The University of Texas at Arlington, Arlington, TX.

Maria Antoniak and David Mimno. 2021. Bad seeds: Evaluating lexical methods for bias measurement. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1889–1904. Association for Computational Linguistics.

James Baker. 2019. Split intransitivity in English. *English Language and Linguistics*, 23:557–589.

Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36:673–721.

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*, 33:130–174.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. *Advances in neural information processing systems*, 29.

Luigi Burzio. 1981. *Intranitive Verbs and Italian Auxiliaries*. Ph.D. thesis, MIT, Cambridge, MA.

Luigi Burzio. 1986. *Italian Syntax: A Government and Binding Approach*. D. Reidel, Dordrecht.

Emmanuele Chersoni, Enrico Santus, Chu-Ren Huang, and Alessandro Lenci. 2021. Decoding word embeddings with brain-based semantic features. *Computational Linguistics*, 47:663–698.

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.

David Dowty. 1991. Thematic proto-roles and argument selection. *Language*, 67:547–619.

Joshua Engels, Isaac Liao, Eric J Michaud, Wes Gurnee, and Max Tegmark. 2025. Not all language model features are linear. In *Proceedings of ICLR*.

Katrin Erk and Marianna Apidianaki. 2024. Adjusting interpretable dimensions in embedding space with human judgments. 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 2675–2686, Mexico City, Mexico. Association for Computational Linguistics.

Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. 2019. Understanding undesirable word embedding associations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1696–1705.

Luana Făgărăşan, Eva Maria Vecchi, and Stephen Clark. 2015. From distributional semantics to feature norms: grounding semantic models in human perceptual data. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 52–57, London.

Aina Garí Soler and Marianna Apidianaki. 2020. BERT knows Punta Cana is not just beautiful, it's gorgeous: Ranking scalar adjectives with contextualised representations. In *Proceedings of EMNLP*, pages 7371–7385.

Gabriel Grand, Idan Asher Blank, Francisco Pereira, and Evelina Fedorenko. 2022. Semantic projection recovers rich human knowledge of multiple object features from word embeddings. *Nature Human Behaviour*, 6:975–987.

Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. 2015. Distributional vectors encode referential attributes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 12–21, Lisbon, Portugal. Association for Computational Linguistics.

Aurélie Herbelot and Eva Maria Vecchi. 2015. Building a shared world: Mapping distributional to model-theoretic semantic spaces. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 22–32.

Yujing Huang. 2018. *Linking form to meaning: Reevaluating the evidence for the unaccusative hypothesis*. Ph.D. thesis, Harvard University, Cambridge, MA.

Ray Jackendoff. 1983. *Semantics and Cognition*. MIT Press, Cambridge, MA.

Brendan T Johns and Michael N Jones. 2012. Perceptual inference through global lexical similarity. *Topics in Cognitive Science*, 4:103–120.

Songhee Kim, Jeffrey R Binder, Colin Humphries, and Lisa L Conant. 2024. Decomposing unaccusativity: a statistical modelling approach. *Language, Cognition and Neuroscience*, 39:1189–1211.

Austin C. Kozlowski, Matt Taddy, and James A. Evans. 2019. The geometry of culture: Analyzing the meanings of class through word embeddings. *American Sociological Review*, 84:905–949.

Beth Levin. 1993. *English verb classes and alternations: A preliminary investigation*. University of Chicago Press, Chicago, IL.

Beth Levin and Malka Rappaport Hovav. 1995. *Unaccusativity. At the Syntax-Lexical Semantics Interface*. MIT Press, Cambridge, MA.

Li Lucy, Divya Tadimeti, and David Bamman. 2022. Discovering differences in the representation of people using contextualized semantic axes. In *Proceedings of EMNLP*, pages 3477–3494.

Mojtaba Nayyeri, Xiaotian Zhou, Sahar Vahdati, Hamed Shariat Yazdi, and Jens Lehmann. 2019. Adaptive margin ranking loss for knowledge graph embeddings via a correntropy objective function.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *Proceedings of ICML*, pages 39643–39666.

David Perlmutter. 1978. Impersonal passives and the Unaccusative Hypothesis. *Papers from the Annual Meeting of the Berkeley Linguistic Society*, 4:157–189.

Lilia Rissman and Asifa Majid. 2019. Thematic roles: Core knowledge or linguistic construct? *Psychonomic Bulletin & Review*, 26:1850–1869.

Alex Rosenfeld and Katrin Erk. 2023. An analysis of property inference methods. *Natural Language Engineering*, 29:201–227.

Dana Rubinstein, Effi Levi, Roy Schwartz, and Ari Rappoport. 2015. How well do distributional models capture different types of semantic knowledge? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short papers)*, pages 726–730.

Antonella Sorace. 2000. Gradients in auxiliary selection with intransitive verbs. *Language*, 76:859–890.

Antonella Sorace. 2004. Gradience at the lexicon-syntax interface: Evidence from auxiliary selection and implications for unaccusativity. In Artemis Alexiadou, Elena Anagnostopoulou, and Martin Everaert, editors, *The Unaccusativity Puzzle*, pages 243–268. Oxford UP, Oxford.

Antonella Sorace. 2011. Gradience in split intransitivity: the end of the Unaccusative Hypothesis? *Archivio Glottologico Italiano*, XCVI(1):67–86.

Akira Utsumi. 2020. Exploring what is encoded in distributional word vectors: A neurobiologically motivated analysis. *Cognitive Science*, 44:e12844.

Robert D Van Valin Jr. 1990. Semantic parameters of split intransitivity. *Language*, 66:221–260.

Joshua E VanArsdall and Janell R Blunt. 2022. Analyzing the structure of animacy: Exploring relationships among six new animacy and 15 existing normative dimensions for 1,200 concrete nouns. *Memory & Cognition*, 50:997–1012.

Su Wang, Stephen Roller, and Katrin Erk. 2017. Distributional modeling on a diet: One-shot word learning from text only. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 204–213.

Mapping the meaning of Hungarian impulsative constructions

Ágnes Kalivoda
Institute for Lexicology, ELTE Research Centre for Linguistics
kalivoda.agnes@nytud.hu

Robert Malouf
Department of Linguistics and Asian/Middle Eastern Languages, San Diego State University
rmalouf@sdsu.edu

Farrell Ackerman
Department of Linguistics, University of California, San Diego
fackerman@ucsd.edu

1 Introduction

This paper presents a corpus-based investigation of the Hungarian impulsative construction, which uses the deverbal nominalizing suffix *-hAtnék*[1] to derive a noun denoting an immediate, often involuntary urge to perform the action denoted by the verb (Kalivoda et al., 2026).

In its original use, the impulsative construction (Cathcart, 2011) denoted basic physiological impulses, as in (1). These verbs formed a clear semantic cluster grounded in embodied experience, and historically they represent the earliest and most entrenched instances of the construction.

(1) a. *küzd-ött-em az* ***i-hatnék-kal***
fight-PST-1SG the drink-URGE-INS
'I was fighting against an urge to drink'

b. *sír-****hatnék****-om támad-t*
cry-URGE-1SG.POSS arise-3SG.PST
'I got an urge to cry'

However, the construction is now fully productive: speakers freely extend it to increasingly complex and figurative uses (2). The construction has moved beyond its original lexical niche, and it currently functions as a general schematic resource for expressing spontaneous desire or impulse toward any kind of activity.

(2) a. ***gomb-nyomogat-hatnék-ja*** *támad*
button-press-URGE-3SG.POSS arise.3SG
'an urge to press buttons seizes him/her'

b. *rá-jö-tt a* ***tuningol-hatnék***
PV-come-PST.3SG the tune-URGE
he/she got that urge to do some tuning

This trajectory raises a distributional question: is the semantic expansion of the construction reflected in, or constrained by, the statistical associations between the construction and its lexical fillers? To investigate this, we conducted two collostructional analyses on the largest available corpus of Hungarian.

2 Text source

The data used in this study comes from the Hungarian portion of the HPLT Monolingual Datasets 3.0 (Aulamo et al., 2023), a large-scale multilingual corpus covering 198 languages and derived from web crawls spanning 2012 to 2024. The full dataset amounts to approximately 50 terabytes of compressed data, making it one of the largest publicly available multilingual resources to date.

The Hungarian subset totals 731 GB of uncompressed raw text, distributed across 9 files in JSONL format. Each document is accompanied by rich metadata, including web register labels and document quality scores.

3 Data collection

Given the size of the corpus, parsing it in full to retrieve occurrences of a relatively rare construction was not feasible. Instead, we adopted the following workflow:

1. We implemented a fast, parallelized keyword-in-context (KWIC) extractor that distributes work across CPU cores. Each worker parses JSONL documents and applies user-defined regular expressions to the text field, extracting each match together with a 200-character context window. The patterns are designed to accommodate the wide range of suffixes that can attach to *-hAtnék* nouns in agglutinative Hungarian.

2. We parsed the resulting hits and their contexts using HuSpaCy (Orosz et al., 2023) to obtain dependency structures.

[1]The suffix has two allomorphs *-hatnék* and *-hetnék* distributed following the rules of vowel harmony. Upper case vowel *A* reflects vowel harmonic allophony.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 291–293
July 3-4, 2026 ©2026 Association for Computational Linguistics

3. We queried these dependency parses with the Treesearch (Malouf, 2025) tool to identify impulsatives and their head predicates.

4. We obtained detailed morphological analyses of the impulsatives using emMorph (Novák et al., 2016), separating the root verb from the derivational *-hAtnék* suffix.

5. For the planned collostructional analyses, we also needed corpus-wide frequency counts for each root verb and head predicate. We generated all possible inflected forms of these items using Hunmorph-foma[2] and retrieved their frequencies via string search over the full corpus.

The resulting dataset consists of 13,526 records, each annotated with linguistic features and metadata. For the present study, the most relevant variables are the root verbs inside the impulsative construction and the predicates co-occurring with *-hAtnék* nouns, along with their frequencies both within the construction and across the corpus as a whole.

4 Data analysis

To examine the semantic organization and usage tendencies of the *-hAtnék* construction, we conducted two collostructional analyses (Stefanowitsch and Gries, 2003; Schmid and Küchenhoff, 2013; Perek, 2015). Constructions are pairings of form and meaning. For simple lexical constructions, this is straightforward. For highly schematic constructions (e.g., the English double object construction) it is more difficult to isolate the contribution of the schema. We can approach this problem from two directions: any lexical items that fill a slot in a construction schema must have a meaning that is compatible with the meaning of the schema; and, a schema gains its meaning from the lexical items that tend to fill its slots (Hilpert, 2006).

Collostructional analysis uses measures of association to identify lexical items that fill slots in a construction with greater than chance frequency. Specifically, to compute the association between the verb *iszik* 'drink' and the impulsative construction, we would set up a 2×2 table:

	-hAtnék	¬*-hAtnék*	Total
iszik	a	b	$a+b$
¬*iszik*	c	d	$c+d$
Total	$a+c$	$b+d$	n

Here a is the frequency of *iszik* in this construction, $a+b$ is the total frequency of all forms of *iszik*, $a+c$ is the total frequency of all instances of the impulsative, and n is the total size of the corpus. The collostructional strength of this association is the degree to which these counts deviate from what we would expect if verb choices and construction choice were statistically independent.

5 Results

The first analysis investigated which verbs are most strongly associated with the [V-*hAtnék*]$_{\text{N}}$ configuration, that is, which verbal bases are most strongly attracted by the -hAtnék derivation.[3].

These verbs pattern into the following semantic fields, each of these illustrated with a few examples:

- bodily functions and expulsion: *okád, hány* ('vomit'), *pisil, vizel* ('urinate'), *köp* ('spit')
- emotional/expressive release: *nevet* ('laugh'), *sír* ('cry'), *röhög, kacag* ('laugh/giggle'), *üvölt* ('howl')
- ingestion: *eszik, zabál* ('eat/gorge'), *iszik* ('drink'), *cigiz* ('smoke')
- motion and escape: *megy* ('go'), *menekül* ('flee'), *szökik* ('escape'), *fut* ('run'), *csavarog* ('roam')
- speech and self-display: *mesél* ('tell'), *beszél* ('speak'), *káromkodik* ('curse'), *dicsekszik* ('boast'), *kérkedik* ('brag')

The construction thus typically selects for verbs denoting barely-controllable or uncontrollable bodily, emotional, or social impulses.

The second analysis examined main verbs that co-occur in the [V-*hAtnék*]$_{\text{N}}$ + V pattern. The most strongly attracted verbs form a remarkably coherent semantic class. The construction overwhelmingly recruits verbs that frame the urge as an external force impinging on the experiencer: *támad* ('attack, arise'), *rájön* ('come over someone'),

[2]https://github.com/r0ller/hunmorph-foma

[3]Only collexemes attracted at the highest level of significance ($p < .0001$, log-likelihood $\geq$ ca. 20) are discussed here.

rátör (‘break upon’), *elfog* (‘seize’), *kitör* (‘erupt’), *környékez* (‘approach’), and *bizsereg* (‘tingle’). A smaller cluster encodes the suppression or fading of the urge: *elfojt* (‘suppress’), *visszafog* (‘hold back’), *elmúlik* (‘pass’), *elpárolog* (‘evaporate’), *erőt vesz* (‘overcome’). This supports the interpretation of the construction as profiling an internal, temporally bounded impulse.

6 Conclusion

The impulsative construction illustrates how expressive language and more specifically constructions that encode internal states, affect, or evaluations, can contribute to constructional productivity and reorganization (Gyselinck, 2018). In the case of the impulsative, expressive meaning appears to facilitate both semantic extension and the construction’s integration into larger patterns of clause-level organization.

This case contributes to ongoing work on how constructions can be characterized and compared in distributional terms, and raises questions about how semantic coherence is maintained as constructions become more schematic and productive.

Acknowledgements

Ágnes Kalivoda’s research has been supported by the OTKA PD project No. 142317 funded by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the PD 22 funding scheme.

References

Mikko Aulamo, Nikolay Bogoychev, Shaoxiong Ji, Graeme Nail, Gema Ramírez-Sánchez, Jörg Tiedemann, Jelmer van der Linde, and Jaume Zaragoza. 2023. HPLT: High performance language technologies. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 517–518, Tampere, Finland. European Association for Machine Translation.

MaryEllen Cathcart. 2011. *Impulsatives: The syntax and semantics of involuntary desire*. Ph.D. thesis, University of Delaware.

Emmeline Gyselinck. 2018. *The role of expressivity and productivity in (re)shaping the constructional network*. Ph.D. thesis, Ghent University.

Martin Hilpert. 2006. Distinctive collexeme analysis and diachrony. *Corpus Linguistics and Linguistic Theory*, 2(2):243–256.

Ágnes Kalivoda, Farrell Ackerman, and Robert Malouf. 2026. Morphological change as systemically motivated bricolage: Hungarian impulsative constructions. *Morphology*, 36(4):1–33.

Robert Malouf. 2025. Treesearch: Pattern matching for dependency treebanks. https://github.com/rmalouf/treesearch.

Attila Novák, Borbála Siklósi, and Csaba Oravecz. 2016. A new integrated open-source morphological analyzer for Hungarian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

György Orosz, Gergő Szabó, Péter Berkecz, Zsolt Szántó, and Richárd Farkas. 2023. Advancing Hungarian Text Processing with HuSpaCy: Efficient and Accurate NLP Pipelines. In *Text, Speech, and Dialogue*, pages 58–69, Cham. Springer Nature Switzerland.

Florent Perek. 2015. *Argument Structure in Usage-Based Construction Grammar*. John Benjamins, Amsterdam/Philadelphia.

Hans-Jörg Schmid and Helmut Küchenhoff. 2013. Collostructional analysis and other ways of measuring lexicogrammatical attraction. *Cognitive Linguistics*, 24(3):531–577.

Anatol Stefanowitsch and Stefan Th. Gries. 2003. Collostructions: Investigating the interaction of words and constructions. *International Journal of Corpus Linguistics*, 8(2):209–243.

Efficient universal generation in a fragment of Optimality Theory

Paul Siewert
Jesus College, University of Cambridge
pks50@cam.ac.uk

Abstract

Various work in computational phonology has studied the computational properties of Optimality Theory. Some algorithms exist for the universal generation problem, including those of Ellison and Tesar, but their domain of applicability is poorly understood. I propose and study a concrete 'minimal' fragment of finite-state Optimality Theory. I show that the universal generation problem for it is efficiently solvable by improving Ellison's Algorithm, demonstrate that it has been implicitly used in the literature, and discuss its limitations. The minimal fragment is a natural and foundational step towards a computationally tractable general formalism for phonological analysis.

1 Introduction

Optimality Theory (OT) (Prince and Smolensky, 1993; Kager, 1999) is a grammatical paradigm providing a template for formal grammars describing phonology. Many formalisms based on OT have been proposed, including but not limited to Ellison (1994), Tesar (1995), and Potts and Pullum (2002), mainly to achieve two purposes. First, different formalisms describe different phenomena in natural language. Some are too restricted to model certain data (Moreton, 1999; Buccola, 2012), while others (or indeed: the same ones in different contexts) have been argued to not be restrictive enough (Potts and Pullum, 2002; Idsardi, 2006; Lamont, 2021).

Second, formalisms differ computationally. The computational problem of determining the predictions of a grammar in some formalism is called the *universal generation problem* for that formalism (Heinz et al., 2009). Several OT formalisms have been shown to have an NP-hard universal generation problem (Eisner, 1997a, 2000; Idsardi, 2006). This means that an efficient general algorithm is likely impossible.

The two aspects described above are independent: It may be that some formalism describes only very simple (e. g. sub-regular) grammars, but does so in a very inefficient manner, so that the universal generation problem is computationally difficult. The formalism would then undergenerate while being intractable. Conversely, a formalism might describe some unnaturally complex grammatical patterns (e. g. cross-serial dependencies), but do so in a straightforward manner so that the universal generation problem is tractable. The formalism would then overgenerate, describing unattested grammars, but it would at least be possible to determine *which* unattested grammar is described.

Computationally tractable models of grammar are desirable for many reasons. First, if phonology is understood as a computational system, to study the algorithmic structure of various formalisms is precisely to study the structure of possible phonologies. Studying the efficiency of a formalism gives insight into its computational structure: In OT, phonological phenomena arise through constraint interaction, and giving an efficient way to determine the output of an OT grammar means understanding the ways in which constraints *can* interact. Second, phonological processing is simple for the human brain. Therefore computational approaches that are even mildly cognitively oriented will naturally aim to implement the necessary operations in a way that is tractable (Ferreira, 2005). Third, our account of a language should not be more complex than the data themselves. In particular, the encoding of a grammar in a framework should not be more computationally complex than what the phenomenon demands. Finally, tractable grammars allow for data science approaches to phonology (Heinz and Idsardi, 2017). Given a sufficiently practical grammatical theory, data can be processed automatically to infer descriptions, and generation and production can then be simulated based on these descriptions. All these arguments motivate both work on heuristic practical algorithms and work towards theoretical guarantees for algorithms.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 294–303
July 3-4, 2026 ©2026 Association for Computational Linguistics

There is no computationally tractable OT-based formalism that encompasses a large number of phonological analyses from the literature, see section 3. On the other hand, the universal generation problem is tractable in practice (Riggle, 2004).

The desideratum of computational feasibility restricts OT fundamentally, so to achieve a tractable formalism it is reasonable to proceed 'from the ground up', incorporating more and more phenomena while staying computationally tractable. Accordingly, this work proposes a formalisation of OT that is limited enough to have a tractable universal generation problem while still containing some of the key structure of OT, termed the 'minimal OT fragment' (mOT). This approach foregrounds the issue of avoiding computational intractability over that of avoiding overgeneration. The reason for this is that overgeneration is easier to address by a later revision of the framework that restricts the allowed grammars, as is done in Lexical Functional Grammar (e. g. Wedekind and Kaplan, 2020).

I first briefly describe OT in section 2 and then review existing work on the generation problem. The focus is on finite-state OT, and Ellison's Algorithm is particularly important. I then introduce mOT in section 4 and describe an algorithm for the universal generation problem. In section 5, I discuss the descriptive capability of mOT, describing both simple extensions and limitations. The results are summarised and discussed in section 6.

2 Optimality Theory

Most formal analyses of phonological phenomena are carried out in one of various frameworks which operate through *constraint optimisation*. Here I consider a flavour of this approach I refer to as Optimality Theory (OT) for convenience. This is consistent with the usage of Prince and Smolensky (1993) who introduced OT. It also includes many but not all more recent variants.

2.1 The general OT framework

OT is based on the observation that most generalisations about natural language have exceptions. It does not describe phonological processes by rules which apply throughout the whole language and in the same way to all forms. Instead, generalisations are described as *filters* which proper surface forms have to pass, called ''constraints''. Starting from a set of feasible phonological representations, called the ''candidates'', these filters are applied successively in a certain order, eliminating some of the candidates in each step. Once only one candidate is left, this is the output of the grammar, often called the ''optimal candidate''. Crucially, constraints can be 'violated', i. e. fail to apply, *as a last resort*: If some constraint would rule out *all* remaining candidates, it is simply ignored from the analysis.

Both rule-based analyses and OT analyses take the underlying representation (UR) as input and determine the SR by simple successive computational steps. However, the former build up and correct the form from the UR while OT narrows down the set of candidates. A given phenomenon may therefore appear very different across frameworks.

The order of constraints usually matters. Almost always the optimal candidate would be ruled out by some of the constraints, i. e. most SRs violate some tendency in the language. Consider two abstract candidates and and two constraints $_1$ and $_2$. Suppose $_1$ rules out but not and $_2$ rules out but not . If $_1$ comes first, only is left for $_2$, which would rule out , but since only is left, is optimal. If instead constraint $_2$ came before $_1$, the reverse would be true, and would be optimal.

The concept of OT becomes a grammatical formalism once we specify the initial set of candidates (representations, GEN in the literature) and which constraints are allowed (EVAL). Various kinds of representations and constraints have been considered in the literature. In this work, the UR is a string of phonemes and the candidates are strings of symbols together with a suitable relation of how the symbols correspond to the UR (McCarthy and Prince, 1993, cf.). This is useful, though most phonology now uses other representations including tiers and/or autosegments. The symbols may represent fully specified phones or suprasegmental marks (e. g. a dot for a syllable boundary).

Most constraints can be understood as penalising certain properties of a candidate. They either just filter out all the candidates that have a certain property (Boolean constraint), or they count the degree or amount of the property and leave only the candidates that have the minimal amount (counting constraint). Most kinds of constraints have a Boolean and a counting version.

2.2 Some constraint types

Many constraints assign violations for co-occurrence of certain features. In particular, if is any string of phones, such as [ɓ] or [spl], the *substring constraint* * assigns one violation for

each occurrence of the substring in a candidate. Such constraints penalise marked phones and phone clusters (typically with of length 2). Often they are also phrased with what I will call *phonological sets* (Chomsky and Halle, 1968; Archangeli and Pulleyblank, 2022) such as S, P, and L standing for any sibilant, plosive, and liquid respectively. The constraint *SPL penalises any sequence of such sounds in that order, and *PP penalises two consecutive plosives. With very broad sets, say denoting the set of all sequences of phones (not standard notation), we can formulate *self-conjunction constraints* (Idsardi, 2006) like *d d. This constraint penalises candidates that contain multiple occurrences of [d].

Constraints on the relation between UR and SR are called *faithfulness constraints*. Typical of this are *identity constraints*, which are violated whenever an underlying feature is not realised with the same value. For instance, the violation count of IDENT(voice) is the number of mismatches between the underlying and surface voicing of consonants. Another type of faithfulness constraints are *parsing constraints*, of which there are two kinds: MAX() for a phoneme and DEP() for a symbol . The former assigns one violation for each elision of and the latter one violation for each epenthesis of .

3 Previous work

There has been substantial interest in the universal generation problem, but the only formalisation of OT that is known to be computationally tractable is that presented in Bruce Tesar's thesis and its extension, see 3.2 below.

3.1 Finite-state OT and Ellison's Algorithm

Ellison (1994) gave the first formalisation of OT. The representations and constraints are conceptually as in section 2.1. The counting function is defined by a non-deterministic finite-state transducer (NFT) reading the UR and the candidate string and writing some number of a symbol called the *violation mark*. This output codes the number of violations in unary; only the smallest possible output counts. The resulting line of work is *finite-state OT*, and the definitive text on this remains Riggle (2004).

Ellison (1994) observed that one can construct what I will call a 'combined evaluator', a transducer outputting multiple numbers that evaluates multiple constraints at once. This is done by adapting the product automaton construction (Rabin and Scott, 1959) to transducers: If the NFTs $_1$, , compute the partial functions $_1$, , from words to numbers counting the violations of some constraints, then the product NFT computes the partial function from words to sequences of numbers defined by () = (()) (on the intersection of the domains). It does so by writing on several output tapes, one for each constraint. Every edge is labelled by the number of violations of each individual constraint incurred by it. The output is an ordered list of natural numbers, the *harmony vector*, in which the -th entry is the violation count for the -th constraint. All should only have one accepting state, and all their states need to have a loop labelled with the empty string .

The combined evaluator gives a very concrete computational view of optimality. To determine which of two candidates is optimal, we compare their harmony vectors entry by entry in the order of the constraint ranking. If both have the same number of violations for one constraint, they are either both eliminated by the constraint, or both pass the constraint. So the decisive constraint is the first constraint where the counts differ. The candidate with more violations for that constraint is worse, and the later constraints do not matter. In mathematical terminology, we compare the harmony vectors *lexicographically*. Thus the function defined by the constraint ranking is equivalently defined by finding the string for which the combined evaluator has the smallest output (in lexicographic ordering).

This view has two theoretical uses: First, it shows that the functions *defined* by this formalism are not arbitrary, but fairly restricted. Using this, Frank and Satta (1998) show that if the constraints are Boolean, the function defined is rational. Second, it gives a mechanism for *computing* the function from the constraint ranking (Ellison, 1994): An optimal candidate can be found by using Dijkstra's Algorithm (Dijkstra, 1959; Baras and Theodorakopoulos, 2010, p. 22) on the combined evaluator. One can construct an automaton recognising *all* optimal candidates by the well-known method that finds all shortest paths from a source in a directed graph, which is to remove all edges whose endpoint is closer to the source than their startpoint. Combined with the product construction above, these steps form *Ellison's Algorithm* for the universal generation problem in finite-state OT. This is the basis for many following algorithms including the one presented in this paper.

Ellison's Algorithm is not efficient in general be-

cause the combined evaluator may be large. Even a binary substring constraint is encoded with 3 states, so for of these constraints the product construction produces a transducer of size 3. A basic but useful heuristic remedy is to simplify the automaton obtained in each step without changing the function it defines. The optimisations of Riggle (2004) of this idea tend to lead to reasonably small evaluation automata for which the optimal candidate can be determined very quickly.

Riggle (2004) argues that in practice the size of the automata is unlikely to grow very large as it ''cannot exceed the number of unique phonological environments to which the grammar is sensitive'' (p. 212, see also section 5.2 there). This is plausible, though not a formal result. The argument is echoed by Kornai (2006) and Kornai (2009), observing that the instances of computationally difficult problems in phonology are very small. However, the argument also acknowledges that there has to be a reason (what Kornai calls a 'guard') why the problems encountered in phonology stay below the theoretically possible complexity. The existence of efficient algorithms would affirm this suspicion and provide such a guard.

Eisner (2000) shows that the universal generation problem in Boolean finite-state OT where candidate length is polynomially bounded is intractable. The argument was simplified by Idsardi (2006) and varied to consider the case where candidate length is unrestricted by Hao (2024). The latter work also formally identifies the exponential size of the product automaton as the 'reason' for the computational complexity. For an algorithm to possibly have a theoretical guarantee of efficiency, the formalism needs to be restricted to very simple constraints.

3.2 Other algorithms

Tesar (1995) describes an algorithm for the universal generation problem that builds the optimal candidate incrementally while reading the UR (loc. cit. p. 16). It is explained how to implement this with an example, and it is shown that this extends to 'local constraints which ban overparsing cycles'. Tesar (1996) extends the algorithm to certain tree representations. I suspect that the very technical nature of the material has obscured its implications.

A different implementation of finite-state OT in *foma* is given by Gerdemann and Hulden (2012), which is likewise practically useful. Karttunen (1998) implements a finite-state OT similar to that of Tesar (1995) in the Xerox calculus. Both consider a handful of concrete constraints for illustrative purposes. Biró (2006) explores a heuristic technique.

3.3 Other formalisms

Two other notable formalisms are primitive OT (Eisner, 1997a) and OT with local tree predicates (Potts and Pullum, 2002). The former uses autosegmental representations and simple constraints for relating different tiers. In the latter, representations are trees and constraints are formulas in a modal logic. Both of these frameworks are illustrated in the respective work to be capable of encompassing large parts of the OT literature. Eisner (1997b) shows how over 120 constraints from the literature can be easily implemented in primitive OT.

Eisner (1997a) observes that with tiers, instances of the Hamilton Path Problem with nodes can be encoded in primitive OT. So even few tiers and simple constraint rankings can result in problems that take a long time to solve. Computation does not play a role for the argument in Potts and Pullum (2002). The logic proposed is trivially computationally intractable, as the case of structures with just one element and a large set of predicates is the Satisfiability Problem. Limiting the formulas to a type for which the satisfiability problem is tractable, such as Horn formulas (Dowling and Gallier, 1984), might yield a more tractable formalism.

Emergent Phonology (Archangeli and Pulleyblank, 2022) is a theory of learning with some computational implications. The fundamental unit of grammar are not segments, but morphologically motivated chunks of phones stored by all possible ways they can be realised. This greatly simplifies computation in practice. However, the computation still needs to consider hundreds or thousands of candidates for any utterance. More importantly, this approach shifts computational complexity to the acquisition of the lexicon. While this is sensible, as learning only has to be performed once, it also shifts the need for computational explanation to the learning phase. Insofar as phonological phenomena apply throughout the language, if a speaker is presented with a new lexical item, they need to infer the corresponding chunks by analogy with other items. To do this, they need to precompute exactly the steps which would need to be performed in other versions of OT.

Lamont (2025) formalises a mechanism called *lexical insertion* in OT, and shows that the universal generation problem becomes *impossible*.

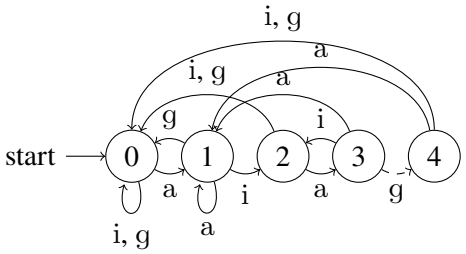


Figure 1: Transducer for *aiag. The states are labelled by the length of the corresponding prefix. The dashed edges assigns a violation.

4 Proposed formalism and algorithm

4.1 The minimal OT fragment

I propose a fragment of finite-state OT with limited representations and constraints. Fix an alphabet , consisting of all phonological 'units' relevant in some languages to be described (including phones, phonemes, and prosodic marks), though I make no mathematical restriction on it. Any string over this alphabet is admissable as a UR. The candidates for a UR are all strings over the alphabet together with any non-crossing relation to the UR that relates at most one segment to each segment. The relation indicates which underlying units are changed, deleted, or inserted on the surface, and will be encoded somewhat abstractly (see below). Only three kinds of constraints are admitted: substring constraints without phonological sets, 'segmental correspondence constraints', and parsing constraints against deletion and epenthesis. These constraints I will refer to as *strictly local constraints* because they only depend on small regions of the surface. An appropriate formalisation of OT will almost certainly be able to describe anything described by interactions of strictly local constraints. In this sense, the proposed formalism is the 'minimal fragment' of OT (mOT). This section shows that the universal generation problem in mOT can be solved by a simple practical algorithm.

mOT is a fragment of finite-state OT in that all the constraints can be encoded as finite-state transducers. The constraint * against the substring of length can be encoded by storing the last characters of the input (requiring || states), and assigning violations along all edges which go to the state corresponding to . Much fewer states can suffice (see fig. 1 and below).

By *segmental correspondence constraints* I mean a generalisation of identity constraints in which the UR is a specific phoneme: The constraint KEEP(,) for some and assigns a violation for every underlying that surfaces as some . Note that KEEP(,) is a formalisation of a ∗MAP-constraint as in Zuraw (2007). These and parsing constraints can be encoded as NFTs with only a single state which read in the UR and the candidate, as is done in Riggle (2004). This has the advantage that the product construction does not increase the size of the automata.

I will use different encodings of the constraints to be described below that are more concise and help with interaction. The automata will be nondeterministic to allow for parsing a candidate like [ta] on UR /tr/ either as epenthesis of [a] and deletion of /r/ or as a change from /r/ to [a]. Thus, formally, the candidates will be all paths in the automaton to be described. Note that I make no restriction that representations or constraints be natural in the phonological sense, so constraints like *ma or KEEP(a, {k}) are allowed.

The algorithm for universal generation in mOT adds a precomputation step to Ellison's Algorithm. Ellison's Algorithm only fails to be efficient because the automaton used to evaluate the constraints may be very large, while the later processing is applicable to any transducer that yields outputs in some totally ordered set. In mOT, inefficiency is avoided by producing an NFT with few edges and states which evaluates any set of mOT constraints at once (computing the harmony vector). Note that this NFT is independent of the ranking of the constraints. The ranking only matters for comparing harmony vectors in order to determe the optimal candidates *from* the NFT.

4.2 Efficient evaluation

For computational purposes, I view faithfulness constraints as an NFT that only reads one input but depends on the UR. This is inefficient for only a single faithfulness constraint, but it makes evaluating multiple constraints simpler.

Suppose some faithfulness constraints and the UR are given. The combined evaluator (see fig. 2 for an example) will have one state for each prefix of , with the empty prefix initial and the full prefix accepting. Intuitively, a state corresponds to how much of has been compared to the input candidate. There are three kinds of edges: (i) for every state and every unit a loop on the state labelled (ii) for every state and every unit an edge to the state that corresponds to the prefix that is one longer, la-

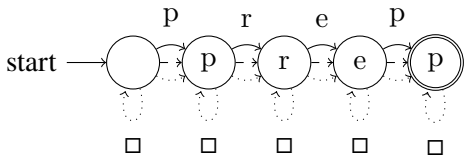


Figure 2: Sketch of evaluator for a set of faithfulness constraints on UR /prep/. Every path is a candidate, but each edge may incur violations of some constraints, i. e. write violation marks to some output tapes: no violation at top arrows, MAX-violation at bottom arrows, DEP-violation at loops. The dashed arrows represent changes to segments, violations of other faithfulness constraints like KEEP(p, {p, b, f}) or KEEP(e, {e, ə}).

belled (iii) for every state an edge to the state that corresponds to the prefix that is one longer labelled with the empty word (denoted). Each type of edge corresponds to some type of constraint violation. A constraint only has to be considered computationally if it occurs in the constraint ranking.

A type (i) edges means reading an in the candidate without reading a corresponding in the UR. Such a situation is an epenthesis/overparse. The edge assigns one violation of DEP() (recall this means printing one violation mark to the appropriate output tape). Type (ii) edges are the most important. An edge from the prefix to the prefix ( ) that is labelled by some means reading an in the candidate while reading a corresponding in the UR. So this edge assigns one violation of KEEP(,) if (for any). A type (iii) edge means moving one segment further in the UR, say reading , without having any corresponding segment in the candidate. Thus this edge assigns one violation of MAX() (deletion/underparse).

Now I describe how to evaluate multiple substring constraints at once. Suppose we have substring constraints $*_1$, …, *. If all have length at most 2, it suffices to keep track of the last read character (e. g. if the last character was [n] and we transition reading [t], that is a violation of *nt). If they are longer, there is also a small automaton. As a preparation, consider the problem: Can we find a small automaton that accepts exactly the words that satisfy all of these constraints? For this it is useful to consider the complement language, which consists of the words that contain at least one of the as a substring. A small automaton for this problem was found by Aho and Corasick (1975); this is also known as a *prefix tree*. This automaton keeps track of *the longest suffix of the input that is a prefix of at least one* (cf. fig. 1). Thus its set of states is the set of all prefixes of the , and the empty prefix is the initial state. In particular, it has at most $1 + |_1| + +||$ states. If one of the occurs as a substring in the input string, then once the automaton has read the input to the end of that occurrence, the state of the automaton will have as a prefix – the last edge taken is a violation of *. Accordingly, to obtain an NFT counting violations of the constraints from the prefix tree, we make all the edges for which this occurs assign a violation of *. This happens to be a deterministic transducer.

Now we have constructed a small combined faithfulness evaluator and a small combined substring evaluator, two NFTs. To obtain a small NFT that evaluates *all* the constraints, we use the product construction of Ellison (1994). Recall that the result will be an NFT which outputs to various tapes in unary. Note again that these tapes do not have an inherent ordering. We only need to assign and consider the ordering in the last steps. Thus this concludes the description of the algorithm.

4.3 Summary of the procedure

Given a ranking of mOT constraints and a UR , we compute an automaton recognising the optimal candidates. This is achieved by Ellison's Algorithm with preprocessing of constraints, as follows:

1. Identify all the substring constraints and use the Aho–Corasick construction to form an NFT that evaluates them at once. (This is independent of .)

2. Use the method described above to form an NFT that evaluates all the faithfulness constraints with UR at once.

3. Use the product construction on the transducers from (1) and (2).

4. Use Dijkstra's Algorithm to determine the harmony vector of each state in the NFT from (3) based on the ranking.

5. Prune all non-optimal edges of the NFT.

Let be the number of constraints, the number of vertices in the combined evaluator, and $= (|| + 1)$ the number of edges. The runtime of the first three steps is (). The runtime of Dijkstra's Algorithm depends on the implementation. Two basic implementations have runtimes $(+ ^2)$ and $(\log + \log)$

respectively. Note that is at most || + 1 times the sum of lengths of the penalised substrings.

Given this structure, one could extend the algorithm to a small number of further constraint types, as long as for each of them *individually* multiple constraints can be evaluated efficiently at once. For example, a regular position structure grammar (Tesar, 1995) can be enforced by adding a *singular* arbitrary Boolean constraint to the ranking.

The combined faithfulness transducer is acyclic. Hence when running Dijkstra's Algorithm on the product transducer, fewer states need to be considered in each step, improving the runtime.

With these modifications, the procedure is identical to that of Tesar (1995), apart from our use of substring constraints with more than two segments.

5 The expressive power of mOT

Many standard OT analyses can be carried out in mOT, especially if some mild extensions are admitted (see below). However, there are restrictions to this because the representations and constraints are limited. Even phenomena that can be described using some standard OT-based formalisms may be not straightforward in mOT. Specifically, mOT lacks two things: It does not allow for phonological sets in the constraints, and its representations do not directly encode prosodic structure. There are workarounds, but how well the second aspect in particular can be addressed may be a subtle question. Being within the framework of finite-state OT, mOT also has some general descriptive limitations. It is difficult to account for cyclic (Moreton, 1999) or opaque phenomena (Buccola, 2012).

mOT also overgenerates. It is generally held that phenomena in natural language phonology can be defined by deterministic finite-state transducers. (Johnson, 1970; Chandlee, 2014). In contrast, Lamont (2021) analyses a hypothetical majority rule pattern in OT. The analysis can be carried out in mOT (more directly using conflation, see section 5.1). The kind of phenomenon is unobserved in natural phonologies, and Lamont observes that it cannot even be generated by *pushdown* transducers.

5.1 Simple extensions of mOT

With some care, mOT constraints can be generalised without impairing the efficiency.

There may be interest in positional faithfulness constraints (Beckman, 1998). To describe a typological difference between Yiddish and English, Wetzels and Mascaró (2001) define the constraint IDONS(voice), which assigns violations if the voice-feature changes, but only if a tautosyllabic sonorant follows. Concretely, this is violated if /zl/ does not surface as [zl]. This behaviour can be achieved in the combined faithfulness evaluator by assigning a violation for this constraint on all edges that change /z/ but only if preceeding /l/. Such constraints can also be used address some cases of opacity (Kager, 1999, pp. 378 ff.).

mOT can also accomodate some common phonological sets, such as in *[+voice]. This constraint behaves differently from the cascade *b ¿ *d ¿ *g, which favours [g] over [b]. It is a *conflation* (de Lacy, 2006, p. 10) of the constraints *b, *d, and *g, in that it is violated if one of these constraints is violated. We can integrate conflation into the combined substring evaluator by having the respective edges write not only to their own tapes but also to a separate tape corresponding to *[+voice]. Faithfulness constraints can likewise be conflated, e. g. the conflation of KEEP(s, {s, z}) and KEEP(z, {s, z}) is violated if an underlying alveolar sibilant does not surface as an alveolar sibilant.

Conflation of multiple constraints involving phonological sets is also possible, like *[+lab]/[+voice], which would be violated by [b], [p], [m], [d], [g]. Presumably [b] would be 'worse' than the others (incurring two violations, so [pak] would win over [bak]). This can be incorporated by having the respective edge write two violations of the constraint.

In all cases, including constraints as described adds no further complexity to the computation besides increasing the number of constraints by 1.

5.2 Prosodic phenomena

Because mOT only uses string representations, care is necessary to represent prosody. The three representational approaches described in Strother-Garcia (2019) can be implemented to an extent: Prosodic boundaries can be encoded by a period or brackets in , the nucleus can be marked similarly (Karttunen, 1998, cf.). In a (C)V(C) language, constraints on the coda and onset are then possible because the shape of the onset and coda is known (they are the sounds that follow and precede the syllable boundary symbol, respectively). However, it is not obvious whether mOT with such encodings of prosody can reproduce analyses which refer to prosodic nodes in more sophisticated ways.

As a positive example, the analysis of the

Hixkaryana stress pattern (Kager, 1999, pp. 150ff.) is in mOT. The alphabet is {., L, H, L, [,]}, where the brackets mark beginning and end of a word. There are some constraints (which I abbreviate SHP) that are never violated, including DEP(L, H, L), MAX, KEEP(H, {H}), IAMB (*.LL, *.HL, *.L., *HH, *.LHL), and NONFINAL (*H]). The constraints *.H and KEEP(L) KEEP(L, {L, H}) are called UNEVENIAMB and PARSE-SYL in Kager (1999).

[HLLLL]		SHP	*.H	KEEP(L)
a.	[.H.LL.LL]	!		
b.	[.HLLLL]			!
c. ☞	[.H.LHLL]			
d.	[.H.LH.HL]		!	

5.3 Tree representations

We can also extend the formalism to allow for context-free position structure grammars and constraints as in Tesar (1996). Thus tree-like prosodic structures can be represented more directly. In step 4 of the algorithm, we then determine an optimal path subject to the constraint that the word is accepted by the grammar. This can be done by dynamic programming following Barrett et al. (2000, sec. 5.3). After pruning the non-optimal edges, the optimal candidates are those that are accepted by the output automaton *and* the grammar. Alternatively, this can be viewed as replacing the finite-state automata by stack automata.

6 Discussion

Previous work on universal generation in OT has found that in practice, finite-state OT is often computationally simple. Theoretically, it is known that even Boolean finite-state OT is computationally intractable in general. Some algorithms are known, but they are limited or poorly understood.

The *minimal OT fragment* (mOT) presented here is an explicit OT formalism proven to be computationally tractable. In fact, the algorithm I present is a concrete addition to standard finite-state technology, so this improvement is not only a theoretical point of asymptotic complexity. It also is the first *positive* computational result on OT since Tesar's Algorithm (Tesar, 1995, 1996) and properly subsumes it by incorporating the Aho–Corassick Algorithm. The algorithm presented is near identical to Tesar's, though I hope this independent, more conceptual approach will be more useful.

The algorithm for mOT extends Ellison's Algorithm (Ellison, 1994) by a precomputation step to deal with specific constraint types. It divides complexity between the transducers and the pre- and postprocessing. Given a set of substring constraints or a set of faithfulness constraints, there exists a small NFT which evaluates all of them at once, which can be constructed and then analysed efficiently. Because the NFT is computationally simple, optimisers of the function defined by it, which are the forms determined by the grammar, can be computed efficiently. This indicates a first reason why OT is 'guarded' against complexity and why the automata of Riggle (2004) stay small, as analyses carried out in mOT are always tractable.

The limitations of mOT stem from the simplicity of representations. Further research may extend mOT to support full correspondence theory, prosodic, autosegmental, and/or gradient representations. Some care will be necessary to ensure extensions stay tractable and ideally practical. However, the simplicity of the algorithm for mOT suggests that some stronger frameworks may still be tractable. Work such as Boersma and van Leussen (2017) has already shown how to make versions of OT which are more powerful than mOT more efficient. Moreover, there is some leverage in designing efficient algorithms for generation in formalisms: Heinz et al. (2009) argue that efficient algorithms for simpler problems than the universal generation problem may be sufficient. Shifting more computational burden to the learning process, as suggested by Emergent Phonology, may also be viable.

mOT is motivated by theoretical computational considerations to exclude intractable problems. Its aim is not to be more descriptively adequate, theoretically complete, or cognitively plausible than other OT formalisms. Rather, the advantage of mOT is its efficiency, which as a desideratum constrains its structure significantly. Many standard segmental analyses from the literature can be phrased in mOT with minor workarounds.

In conclusion, this work presents a starting point for the broader project of identifying computationally tractable and descriptively adequate versions of OT, which will have numerous theoretical implications and applications.

Acknowledgement. In the making of this paper, four LLM prompts were used to aid literature search and improve style. I thank the three anonymous reviewers, my advisor Fermín Moscoso del Prado Martín, and my friend Yorick Zeschke for their invaluable comments and encouragement.

References

Alfred Vaino Aho and Margaret J. Corasick. 1975. Efficient string matching: an aid to bibliographic search. *Communications of the ACM*, 18(6):333–340.

Diana Archangeli and Douglas Pulleyblank. 2022. *Emergent Phonology*. Language Science Press.

John S. Baras and George Theodorakopoulos. 2010. *Path Problems in Networks*. Springer.

Chris Barrett, Riko Jacob, and Madhav Marathe. 2000. Formal-language-constrained path problems. *SIAM Journal on Computing*, 30(3):809–837.

Jill N Beckman. 1998. *Positional faithfulness*. Ph.D. thesis, University of Massachusetts, Amherst.

Tamás Sándor Biró. 2006. *Finding the right words: Implementing Optimality Theory with simulated annealing*. Ph.D. thesis, Rijksuniversiteit Groningen.

Paul Boersma and Jan-Willem van Leussen. 2017. Efficient evaluation and learning in multilevel parallel constraint grammars. *Linguistic Inquiry*, 48(3):349–388.

Brian Buccola. 2012. On the expressivity of Optimality Theory versus ordered rewrite rules. In *International Conference on Formal Grammar*, pages 142–158. Springer.

Jane Chandlee. 2014. *Strictly local phonological processes*. Ph.D. thesis, University of Delaware, Newark.

Noam Chomsky and Morris Halle. 1968. *The sound pattern of English*. Harper & Row.

Paul de Lacy. 2006. *Markedness: Reduction and preservation in phonology*. Cambridge University Press.

Edsger Wybe Dijkstra. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271.

William F. Dowling and Jean Henri Gallier. 1984. Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *The Journal of Logic Programming*, 1(3):267–284.

Jason Eisner. 1997a. Efficient generation in Primitive Optimality Theory. In *35th annual meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 313–320.

Jason Eisner. 1997b. What constraints should OT allow? Technical report, LSA. ROA 204.

Jason Eisner. 2000. Easy and hard constraint ranking in OT: Algorithms and complexity. In *Proceedings of the Fifth Workshop of the ACL Special Interest Group in Computational Phonology*, pages 22–33.

T Mark Ellison. 1994. Phonological derivation in Optimality Theory. In *COLING 1994 Volume 2: The 15th International Conference on Computational Linguistics*.

Fernanda Ferreira. 2005. Psycholinguistics, formal grammars, and cognitive science. *Linguistic review*, 22.

Robert Frank and Giorgio Satta. 1998. Optimality Theory and the generative complexity of constraint violability. *Computational linguistics*, 24(2):307–315.

Dale Gerdemann and Mans Hulden. 2012. Practical finite state Optimality Theory. In *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*, pages 10–19.

Sophie Hao. 2024. Universal generation for Optimality Theory is PSPACE-complete. *Computational Linguistics*, 50(1):83–117.

Jeffrey Heinz and William James Idsardi. 2017. Computational phonology today. *Phonology*, 34(2):211–219.

Jeffrey Heinz, Gregory M Kobele, and Jason Riggle. 2009. Evaluating the complexity of Optimality Theory. *Linguistic Inquiry*, 40(2):277–288.

William James Idsardi. 2006. A simple proof that Optimality Theory is computationally intractable. *Linguistic Inquiry*, 37(2):271–275.

Charles Douglas Johnson. 1970. *Formal aspects of phonological description*. Ph.D. thesis, University of California, Santa Barbara.

René Kager. 1999. *Optimality Theory*. Cambridge University Press, Cambridge.

Lauri Karttunen. 1998. The proper treatment of Optimality in computational phonology. In *Finite state methods in natural language processing*.

András Kornai. 2006. Guarded optimalism. Technical report, Rutgers Optimality Archive.

András Kornai. 2009. The complexity of phonology. *Linguistic Inquiry*, 40(4):701–712.

Andrew Lamont. 2021. Optimality Theory implements complex functions with simple constraints. *Phonology*, 38(4):729–740.

Andrew Lamont. 2025. Optimality Theory with lexical insertion is not computable. *University of Massachusetts Occasional Papers in Linguistics*, 42.

John Joseph McCarthy and Alan Prince. 1993. *Prosodic Morphology: Constraint interaction and satisfaction*. Rutgers Center for Cognitive Science.

Elliott Moreton. 1999. Non-computable functions in Optimality Theory. In John Joseph McCarthy, editor, *Optimality Theory in Phonology: A Reader*. Blackwell Publishing.

Christopher Potts and Geoffrey Keith Pullum. 2002. Model theory and the content of OT constraints. *Phonology*, 19(3):361–393.

Alan Prince and Paul Smolensky. 1993. Constraint interaction in generative grammar. *Rutgers University Center for Cognitive Science, New Brunswick*.

Michael Oser Rabin and Dana Scott. 1959. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3(2):114–125.

Jason Alan Riggle. 2004. *Generation, recognition, and learning in finite state Optimality Theory*. Ph.D. thesis, University of California, Los Angeles.

Kristina Strother-Garcia. 2019. *Using Model Theory in Phonology: A Novel Characterization of Syllable Structure and Syllabification*. Ph.D. thesis, University of Delaware, Newark.

Bruce Benson Tesar. 1995. *Computational Optimality Theory*. Ph.D. thesis, University of Colorado, Boulder.

Bruce Benson Tesar. 1996. Computing optimal descriptions for Optimality Theory grammars with context-free position structures. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 101–107.

Jürgen Wedekind and Ronald M. Kaplan. 2020. Tractable Lexical-Functional Grammar. *Computational Linguistics*, 46(3):515–569.

Leo Wetzels and Joan Mascaró. 2001. The typology of voicing and devoicing. *Language*, 77(2):207–244.

Kie Zuraw. 2007. The role of phonetic knowledge in phonological patterning: corpus and survey evidence from Tagalog infixation. *Language*, 83(2):277–316.

Learning Process Interaction Through Simplex ISL Transducers

Chenli Wang and Adam Jardine
Rutgers University

Abstract

This paper investigates the learnability of interacting phonological processes by restricting the hypothesis space to a subregular class of functions. Interacting processes can be modeled as function composition, where the output of one function serves as the input to another. We focus specifically on interactions between two simplex Input Strictly Local (ISL_2) functions, a proper subclass of the ISL function class. We propose a decomposition algorithm that reconstructs both the individual component processes and their relative ordering by exploiting structural properties of simplex ISL_2 transducers and their compositions. This work provides an initial step toward understanding how learners can infer not only single phonological processes, but structured interactions between processes.

1 Introduction

This paper is motivated by a fundamental learning problem in natural language phonology: the acquisition of interacting functions. Previous work has established learning procedures for individual functions, including those characterized as subsequential functions (Oncina et al., 1993), and more efficient algorithms for subclasses of subsequential functions such as Input Strictly Local (ISL) and Output Strictly Local (OSL) functions (Chandlee, 2014; Chandlee et al., 2015; Jardine et al., 2014). Additional work has also developed algorithms that simultaneously infer both the mapping and the underlying representations from surface forms alone (Hua and Jardine, 2021).

However, many real-world phonological transformations are not realized by a single process. Phonological processes frequently interact when the application of one process affects the environment required by another, giving rise to both transparent and opaque interactions (Kiparsky, 1973; Baković, 2007). For example, in Votic (Ariste, 1968; Odden, 2005), raising and fronting of word-final mid vowels to [i] feeds palatalization of /k/.

(1) Feeding in Votic (Odden, 2005, p. 100)
a. /kurkə/ → kurki → [kurtʃi] ‘stork’
b. /əlʸkə/ → əlʸki → [əlʸtʃi] ‘straw’

In such cases, the observable mapping corresponds to a composed function: the output of the first function serves as the input to the second. The learners task, then, is to recover the component functions as well as their ordering.

In this work, we investigate the learnability of interacting functions that can be characterized as simplex ISL_2 functions, a subclass of the ISL function class, under subregular constraints. We frame the problem as one of function decomposition. Specifically, we assume that the learner has access to the output of OSTIA (Oncina et al., 1993), which is first used to learn a transducer representing the composed function. This assumption is justified because OSTIA can provably learn any subsequential function, and the class of subsequential functions is closed under composition. We then examine how to systematically decompose a function into its constituent functions.

Focusing on interactions between simplex ISL_2 functions, we introduce the Simplex ISL_2 Function Interaction Decomposition Algorithm (SI_2FIDA), which provably recovers both the individual functions and their relative ordering from the composed transducer. More broadly, this work contributes to our understanding of how locality constraints can be leveraged to learn structured compositions of functions. Indeed, this work generalizes the work of Hua and Jardine (2021) decomposing phonological functions. While Hua and Jardine show how to decompose the composition of a homomorphism and a simplex ISL_2 function, the current work shows how decomposition can be accomplished when both functions are sim-

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 304–312
July 3-4, 2026 ©2026 Association for Computational Linguistics

plex ISL_2 functions.

The paper is structured as follows. Section 2 introduces the relevant notation and background concepts. Section 3 defines the relevant function classes. Section 4 establishes key properties of compositions of two simplex ISL_2 functions. Section 5 formalizes the learning problem. Section 6 presents the algorithm, demonstrates its operation, and proves its correctness, along with its data and time complexity. Section 7 concludes and discusses directions for future work.

2 Preliminaries

Let Σ be a finite alphabet. Σ^* denotes the set of all finite strings over Σ. The empty string is denoted by λ. For strings w and v, both wv and $w \cdot v$ denotes their connatenation. If $w = uv$ then $u^{-1} \cdot w = v$ and $w \cdot v^{-1} = u$. The set of *prefixes* of a string w is defined as: $\text{pref}(w) = \{u \in \Sigma^* \mid (\exists v \in \Sigma^*)[uv = w]\}$. The set of *suffixes* is defined as $\text{suff}(w) = \{v \in \Sigma^* \mid (\exists u \in \Sigma^*)[uv = w]\}$. The *k-prefix* of w is $\text{pref}_k(w) = u$ (respectively $\text{suff}_k(w) = u$) such that u is a prefix (respectively suffix) of w and $|u| = k$ if $|w| > k$, otherwise $u = w$. The notion of prefixes and suffixes can be extended to sets of strings. For example, $\text{pref}(S) = \bigcup_{w \in S} \text{pref}(S)$.

For a pair of strings w and v, let $w \wedge v$ denote their *longest common prefix*, which is a unique u s.t. $u \in \text{prefs}(w) \cap \text{prefs}(v)$ and for all $x \in \text{prefs}(w) \cap \text{prefs}(v), |u| \geq |x|$.

The function $f : \Sigma^* \to \Sigma^*$ maps strings to strings. As we are interested in functions that may be composed in either order, we assume that the input alphabet is the same as the output alphabet. $\text{dom}(f)$ and $\text{ran}(f)$ denote the domain and range of f respectively.

For f, $\mathring{f}(w)$ denotes the common output of w, which is defined as $\mathring{f}(w) \stackrel{\text{def}}{=} \bigwedge_{wu \in \text{dom}(f)} f(wu)$. For any $w \in \Sigma^*$, f_w denotes the tail function of w with respect to f, which is defined as $f_w(u) \stackrel{\text{def}}{=} \mathring{f}(w)^{-1} \cdot f(wu)$. Intuitively, $f_w(u)$ returns what the output of u is after removing the output we can attribute to w. $\mathring{f}_w$ denotes the tail prefix function, which is defined as $\mathring{f}_w \stackrel{\text{def}}{=} \bigwedge_{wu \cdot v \in \text{dom}(f)} f_w(uv)$. Intuitively, $\mathring{f}_w$ returns the common output of wu minus the common output of w.

3 Function Classes

A function f is said to be *(left-)subsequential* iff the set of its unique tail functions, $\{f_w \mid w \in \Sigma^*\}$, is finite (Schützenberger, 1977; Sakarovitch, 2009). A bold capital letter (e.g., $\mathbf{F}$) denotes the set of tail functions of f. Intuitively, each tail function f_w represents the behavior of f after processing some prefix w.

Subsequential functions are exactly those described by *subsequential finite-state transducers* (SFSTs), which are deterministic finite-state transducers with outputs on the states. An SFST T is a tuple $\langle Q, q_0, Q_\ltimes, \delta, \omega, \iota \rangle$, where Q is a finite set of states, $q_0 \in Q$ is the initial state, $Q_\ltimes \subseteq Q$ is the set of final states, $\delta : Q \times \Sigma \to Q$ is the transition function, $\omega : Q \times \Sigma \to \Sigma^*$ is the output function, and $\iota : Q_\ltimes \to \Sigma^*$ is the final-state output function. To define the function described by an SFST, we extend δ and ω to functions δ^* and ω^* on $Q \times \Sigma^*$ as follows: for $w = \lambda$ and any $q \in Q$, $\delta^*(q, \lambda) = q$ and $\omega^*(q, \lambda) = \lambda$; otherwise, for any $\sigma \in \Sigma$, $\delta^*(q, w\sigma) = \delta(\delta^*(q, w), \sigma)$ and $\omega^*(q, w\sigma) = \omega^*(q, w)\omega(\delta^*(q, w), \sigma)$. Then, for an SFST T, the function $f(T) : \Sigma^* \to \Sigma^*$ is defined as follows: $f(w)$ is defined iff $\delta^*(q_0, w) \in Q_\ltimes$, and in that case, $f(w) = \omega^*(q_0, w)\iota(\delta^*(q_0, w))$.

The states in an SFST for f correspond to the tail functions of f, that is, for each $q \in Q$, $f_q = f_w$ for any prefix w that reaches q (i.e., $\delta^*(q_0, w) = q$). An SFST is *onward* if for any $q \in Q$ and $\sigma \in \Sigma$, $\omega(q, \sigma) = \mathring{f}_q(\sigma)$. The *canonical* SFST for f is the minimal onward transducer for f. We can effectively generate the canonical SFST for f from any arbitrary SFST for f (Sakarovitch, 2009).

A more restricted subclass of subsequential functions is known as *Input Strictly Local* (ISL) functions (Chandlee, 2014).

Definition 1 (ISL)**.** A function f is *input strictly k-local (ISL$_k$)* iff for any $u, v \in \Sigma^*$, $\text{suff}_{k-1}(u) = \text{suff}_{k-1}(v)$ implies that $f_u = f_v$.

In other words, two prefixes u and v that share the same last $k-1$ symbols must induce the same tail function. Intuitively, an ISL_k function is one whose computation depends only on the most recent $k-1$ input symbols.

Every ISL function can be represented by a subsequential finite-state transducer whose states correspond to possible $(k-1)$-length suffixes of input strings. Formally, for an ISL_k function f, the set of tail functions is $\mathbf{F} = \{f_w \mid w \in \text{suff}_{k-1}(\Sigma^*)\}$.

That is, each state in the corresponding SFST represents the continuation behavior of f from a given $(k-1)$-symbol input context.

In order to make the learning problem tractable, we focus on ISL_k functions for $k = 2$. Consequently, the behavior of the function following a prefix w depends only on the final symbol of w (or λ if $w = \lambda$).

To further simplify the learning problem, and to model the notion of a single phonological process, we further narrow the focus to a subclass of the ISL_2 functions which make a single change.

Definition 2 (Left-simplex ISL_2 function (Hua and Jardine, 2021))**.** An ISL_2 function f is left-simplex iff it is properly ISL_2 and there is exactly one $f_{env} \in \mathbf{F}$ and $\tau \in \Sigma$ such that $\mathring{f}_{env}(\tau) \neq \tau$.

This class captures functions whose output depends only on the immediately preceding input symbol and permits exactly one symbol to be changed after exactly one class of prefixes. Henceforth we will refer to left-simplex ISL_2 functions simply as simplex ISL_2 functions.

Let e denote the *environment*, or the 1-suffix after which the change in f occurs (i.e., $f_e = f_{\text{env}}$). Likewise, we refer to $\tau \in \Sigma$ such that $f_{\text{env}}(\tau) = w_\tau \neq \tau$ as the target of f, with $w_\tau \in \Sigma^*$ representing the output of τ. Finally, let f_{def} denote the default tail function, which maps every symbol to itself. An example of a transducer that captures a simplex ISL_2 function is given in Fig. 1.

Example 1. Suppose a simplex ISL_2 function f over $\Sigma = \{a, b, c\}$ that maps a to b when it follows c, while all other strings map to themselves. In this case, $e = c$, $\tau = a$, and $w_\tau = b$; i.e. $g_c(a) = b$. Accordingly, $f_{\text{env}} = f_c$, and $f_{\text{def}} = f_\lambda = f_a = f_b$, representing the tail function associated with all other 1-suffixes. Thus, a unique simplex ISL_2 function can be fully specified once e, τ, and w_τ are identified.

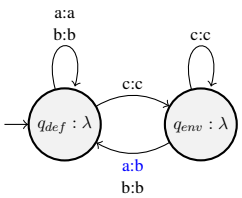


Figure 1: A simplex ISL_2 SFST for f from Ex. 1.

The following remark about the structure of SFSTs for simplex ISL_2 functions straightforwardly follows the from the definition of simplex ISL_2 functions. For a state $q \in Q$ of a transducer T, let its set of *input* 1-*suffixes*

$$\texttt{IS}_1(q) = \texttt{suff}_1(\{w \mid \delta^*(q_0, w) = q\})$$

be the set of 1-suffixes in the input seen before reaching that state.

Remark 1. For the canonical SFST for any simplex ISL_2 function:

1. There are exactly two states $\{q_{\text{def}}, q_{\text{env}}\}$.
2. $|\,\text{IS}_1(q_{\text{env}})| = 1$; if $\text{IS}_1(q_{\text{env}}) = \{\lambda\}$, then $q_{\text{env}} = q_0$ (q_{env} is the initial state). Otherwise, there is exactly one symbol σ such that $\delta(q, \sigma) = q_{\text{env}}$ for any $q \in \{q_{\text{def}}, q_{\text{env}}\}$. Thus $\text{IS}_1(q_{\text{env}}) \cap \text{IS}_1(q_{\text{def}}) = \emptyset$.
3. There is exactly one symbol τ for which $\omega(q_{\text{env}}, \tau) \neq \tau$. For any other state q and symbol σ, $\omega(q, \sigma) = \sigma$.

The SFST in Figure1 is a canonical SFST for the function f in Ex. 1.

4 Composition of input-strictly local functions

4.1 Interacting simplex ISL_2 functions

The primary interest of this paper is the composition of simplex ISL_2 functions. Two functions f and g are said to *interact* when $g(f(x)) \neq f(g(x))$ for at least one x. Below is an example of two interacting functions.

Example 2. Consider two simplex ISL_2 functions, f and g, with tail functions $\mathring{f}_c(a) = b$ and $\mathring{g}_b(d) = b$. These two functions interact because their order of application yields different results. If f is ordered first, it creates the environment for g to apply in the string *cad*, producing the derivation: $cad \xrightarrow{f} cbd \xrightarrow{g} cbb$. In contrast, when g applies first, the input *cad* is mapped to *cbd*.

When both f and g are restricted to simplex ISL_2 functions, their interaction occurs when one function either creates or removes the environment required by the other. This can be formalized in terms of tail functions as follows.

Remark 2. If $g \circ f \neq f \circ g$, then at least one of the following two situations holds:

1. There is some prefix w whose tail function $g_w \in \mathbf{G}$ of g is such that $g_{\mathring{f}(w)} \neq g_w$ or whose tail function $f_w \in \mathbf{F}$ of f is such that

$f_{\mathring{g}(w)} \neq f_w$. That is, either f affects some tail function of g or g affects some tail function of f.

2. There are some w and σ for whom $g_{\mathring{f}(w)}(f_w(\sigma)) \neq f_w(\sigma)$ or $f_{\mathring{g}(w)}(g_w(\sigma)) \neq g_w(\sigma)$. That is, either g 'obscures' an output of f or vice versa.

Example 3. Let f and g be as in Example 2. Because $g(cad) = cad$ (i.e., identity) but $g(f(cad)) = cbb$ (i.e., the change affected by g has occured), this means that the tail function of ca for g is distinct from the tail function of $\mathring{f}(ca) = cb$, as $g_{ca}(d) = d \neq g_{\mathring{f}(ca)}(d) = b$.

Because the present problem concerns learning from the composition of two functions f and g, we must limit ourselves to cases in which each the application of individual function can be observed from their composition. We call this *recoverable*.

Definition 3 (Recoverable)**.** For two functions f and g, we say that each is *recoverable* from $h = g \circ f$ iff there is some x and some y such that $f(x) = h(x)$ and $g(y) = h(y)$.

Non-recoverable cases are exactly those in which both f and g have the same environment and the change in f serves as the target of g, as shown in Example 4.

Example 4 (A non-recoverable function)**.** Consider the simplex ISL_2 functions f and g in which the environment tail function for f is $\mathring{f}_b(a) = c$ and $\mathring{g}_b(c) = d$. In this case, both f and g share the same environment (b) *and* the output of the change of f (c) is exactly the target of g. Thus, while $f(ba) = bc$, $g(f(ba)) = bd$. In fact, there is no string x such that $f(x) = g \circ f(x)$, as $g \circ f$ will change any a following a b directly into a d. Thus, in $g \circ f$ there is no way to observe that f changes a a into a c.

We henceforth are only concerned with interactions in which both f and g are recoverable from $g \circ f$ and $f \circ g$. We can make the following observation, which formalizes the above discussion about the nature of interaction.

Lemma 1. For two interacting simplex ISL_2 functions f and g, let e' and τ' be the environment and target of f, respectively, and let e'' and τ'' be the environment and target for g. Let $s' = \text{suff}_1(\mathring{f}(e'\tau'))$; that is, the 1-suffix after seeing the output of the change in f. Define s'' similarly for g.

If f and g are both recoverable from $g \circ f$ and $f \circ g$, then one of the following is true: 1) $e' = s''$; 2) $e'' = s'$; 3) $e' = \tau''$; or 4) $e'' = \tau'$.

Proof. From Remark 2 and the fact that both f and g are recoverable, if $g \circ f \neq f \circ g$ then it must be the case that there is some w such that either $g_w \neq g_{\mathring{f}(w)}$ or $f_w \neq f_{\mathring{g}(w)}$. (Remark 2-2 cannot be true because if f and g are recoverable, then one cannot obscure the output of the other.)

We focus on the case in which $g_w \neq g_{\mathring{f}(w)}$. Since g is simplex and thus only one non-identity change is allowed, then either $g_w = g_{e''}$ or $g_{\mathring{f}(w)} = g_{e''}$. That is, one must be the environment tail function, as one must have some non-identity output for some input symbol σ. Furthermore, they must disagree on the target τ''. Therefore either $g_w = g_{e''}$ and $g_{e''}(\tau'') \neq g_{\mathring{f}(w)}(\tau'')$ or $g_{\mathring{f}(w)} = g_{e''}$ and $g_w(\tau'') \neq g_{e''}(\tau'')$.

Since f itself is only allowed to change τ' to s' after seeing environment e', then if $w \neq \mathring{f}(w)$ then $e'\tau'$ must be a suffix of w and s' must be a suffix of $\mathring{f}(w)$. Thus if $g_w \neq g_{\mathring{f}(w)}$ either $\tau' = e''$ or $s' = e''$. This establishes cases (2) and (4) above. The same logic applied to the case in which $f_w \neq f_{\mathring{g}(w)}$ establishes cases (1) and (4). □

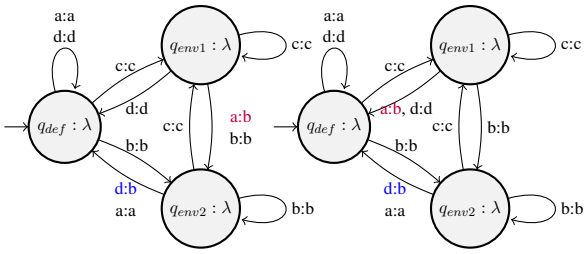
4.2 Interaction and composition of SFSTs

The composition of two subsequential functions can be viewed through the composition operation on their respective SFSTs. For two SFSTs T' and T'', their composition is given by the following from Hua and Jardine (2021), which in turn was adapted from Lothaire (2005).

Definition 4 (Composition of SFSTs)**.** For two SFSTs $T' = \langle Q', q'_0, Q'_\ltimes, \delta', \omega', \iota' \rangle$ and $T'' = \langle Q'', q''_0, Q''_\ltimes, \delta'', \omega'', \iota'' \rangle$, $T'' \circ T' := \langle Q' \times Q'', (q'_0, q''_0), Q'_\ltimes \times Q''_\ltimes, \delta, \omega, \iota \rangle$ where

$$\delta((q', q''), \sigma) := (\delta'(q', \sigma), \delta''^*(q'', \omega'(q', \sigma)));$$
$$\omega((q', q''), \sigma) := \omega''^*(q'', \omega'(q', \sigma)); \text{ and}$$
$$\iota((q'_\ltimes, q''_\ltimes)) := \omega''^*(q''_\ltimes, \iota'(q'_\ltimes)).$$

The compositions $g \circ f$ and $f \circ g$ from Example 3 are shown in Figure 2 as three-state onward SFSTs. Although the construction in Definition 4 yields four states, the fourth state is unreachable from the initial state and is therefore omitted.



(a. $T_{g \circ f}$**)** **(b.** $T_{f \circ g}$**)**

Figure 2: Composition

The following states some important properties of the canonical SFST representing the composition of any two left-simplex ISL_2 functions. Note that any ISL_k class is not closed under composition (Chandlee, 2014) and therefore the composed transducer may not be ISL_2. However, this does not affect the success of our algorithm.

Lemma 2. Let f and g be two simplex ISL_2 functions. Let $\tau' \in \Sigma$ be the one symbol such that $f_{\text{env}}(\tau') \neq \tau'$ and similarly $\tau'' \in \Sigma$ be the one symbol such that $g_{\text{env}}(\tau'') \neq \tau''$. (It may be the case that $\tau' = \tau''$.) The following is true for the canonical SFST for $T_{g \circ f}$:

1. There is exactly one state $q_{\text{def}} \in Q$ for which $\omega(q_{\text{def}}, \sigma) = \sigma$ for all $\sigma \in \Sigma$.

2. Out of the remaining states, there are at most two states that are either the initial state or are reachable from q_{def} by a path one transition long. Call them q_{env1} and q_{env2}(it may be the case that $q_{\text{env1}} = q_{\text{env2}}$).

3. For one $q' \in \{q_{\text{env1}}, q_{\text{env2}}\}$, $\omega(q, \tau') = f_{env}(\tau')$ and similarly for one $q'' \in \{q_{\text{env1}}, q_{\text{env2}}\}$, $\omega(q, \tau') = g_{env}(\tau'')$.

Proof. Let T' be the canonical SFST for f and T'' be the canonical SFST for g. The set of states for T' and T'' are $\{q'_{\text{def}}, q'_{\text{env}}\}$ and $\{q''_{\text{def}}, q''_{\text{env}}\}$, respectively. Thus, by Definition 4 of SFST composition, the set of states for $T'' \circ T'$ is, maximally, $\{(q'_{\text{def}}, q''_{\text{def}}), (q'_{\text{def}}, q''_{\text{env}}), (q'_{\text{env}}, q''_{\text{def}}), (q'_{\text{env}}, q''_{\text{env}})\}$. Lemma 2-1 holds for $(q'_{\text{def}}, q''_{\text{def}})$ and only $(q'_{\text{def}}, q''_{\text{def}})$, as any state corresponding to q'_{env} or q''_{env} will have at least one outgoing non-identity transition.

Lemma 2-2 holds because each of q'_{env} and q''_{env} can be reached from q'_{def} and q''_{def}, respectively, by exactly one 1-suffix. For each of these states, either this 1-suffix is λ (and thus either q'_{env} or q''_{env} is the start state) or it is some σ such that $\delta'(q'_{\text{def}}, \sigma) = q'_{\text{env}}$ or $\delta''(q''_{\text{def}}, \sigma) = q''_{\text{env}}$. Thus, by the definition of composition, $q_{\text{env1}} = (q'_{\text{env}}, q''_{\text{def}})$ and $q_{\text{env2}} = (q'_{\text{def}}, q''_{\text{env}})$ are both reachable from q_{def} by a single transition.

In contrast, if a fourth state $q_{\text{extra}} = (q'_{\text{env}}, q''_{\text{env}})$ exists, the path from q_{def} to q_{extra} is at minimum two transitions. In other words, q_{extra} is only accessible from q_{env1} or q_{env2}. By the definition of simplex ISL_2, there is only one transition from q'_{def} to q'_{env}. Let σ be the single symbol in Σ such that $\delta'(q'_{\text{def}}, \sigma) = q'_{\text{env}}$. By the definition of simplex ISL_2, any transition out of q'_{def} must be an identity transition, so it is also the case that $\omega'(q'_{\text{def}}, \sigma) = \sigma$. Thus, by the definition of composition, $\delta(q_{\text{def}}, \sigma) = \delta((q'_{\text{def}}, q''_{\text{def}}), \sigma) = (\delta'(q'_{\text{def}}, \sigma), \delta''^{*}(q''_{\text{def}}, \sigma)) = (q'_{\text{env}}, q'')$ for some q'' such that either $q'' = q''_{\text{def}}$ or $q'' = q''_{\text{env}}$. If $q'' = q''_{\text{def}}$, then the only transition to $q_{\text{extra}} = (q'_{\text{env}}, q''_{\text{env}})$ must be $\delta((q'_{\text{env}}, q''_{\text{def}}), \sigma)$ and thus any string must first traverse $(q'_{\text{env}}, q''_{\text{def}})$ before reaching q_{env}.

In the other case that $q'' = q''_{\text{env}}$, then both $\delta'(q'_{\text{env}}, \sigma) = q'_{\text{env}}$ and $\delta''(q''_{\text{def}}, \sigma) = q''_{\text{env}}$. However, then for any distinct $\tau \neq \sigma \in \Sigma$, then $\delta'(q'_{\text{env}}, \tau) = q'_{\text{def}}$ and $\delta''(q''_{\text{env}}, \tau) = q''_{\text{def}}$, and so all other transitions in T lead back to q_{def}. Thus if $q'' = q''_{\text{env}}$, there cannot be four distinct states (in fact there can only be two). □

To clarify Lemma 2-3, q_{extra} is a distinct fourth state exactly in cases where both processes must apply, but the non-identity transition exiting the first environment state cannot directly enter the second environment state. In such configurations, the extra state mediates the interaction.

Lemma 3. Assume τ', τ'', T_f, and T_g as in Lemma 2. Let e' be the unique 1-suffix that reaches q'_{env} in T_f and e'' be the unique 1-suffix that reaches q''_{env} in T_g. Let $s' = \text{suff}_1(\omega'^{*}(q'_0, e'\tau'))$; that is, the 1-suffix seen in the output after seeing both the environment and target for T_f. Similarly, let and $s'' = \text{suff}_1(\omega''^{*}(q''_0, e'\tau''))$.

The following hold of s', e', and e''.

1. If $s' = e''$, then $\delta(q_{\text{env1}}, \tau') = q_{\text{env2}}$.

2. If $e' = \tau''$, then $\delta(q_{\text{env2}}, \tau'') = q_{\text{env1}}$.

3. If neither is true, then $f \circ g = g \circ f$.

Proof. Points (1) and (2) follow directly from Lemma 1 and the definition of composition. Point (3) follows from Lemma 1. □

The Algorithm in §6 uses the properties of Lemmas 2 and 3 to identify the individual functions f and g.

5 Learning Paradigm

Informally, the learning goal is to identify two functions, f and g, along with their ordering, from a transducer obtained via OSTIA that represents their composition, under the assumption that each function is a simplex ISL_2 function. Formally, we adopt de la Higuera (2010)'s identification in the limit from polynomial time and data. Under this paradigm, the goal is to identify an algorithm that, given a sufficiently large positive sample for a specified class of functions, can recover a representation of each target function and their ordering in polynomial time. Such a sufficient sample must be of polynomial size relative to the size of the functions representation.

Consider a class $\mathbb{T}$ of functions represented by a class $\mathbb{R}$ of finite representations, there exists a total, surjective mapping $N : \mathbb{R} \to \mathbb{T}$ from representations $r \in \mathbb{R}$ to functions in $\mathbb{T}$.

Definition 5 (Sample)**.** For a function $t \in \mathbb{T}$, a sample S of t is a finite set of pairs (w, v) such that $t(w) = v$ for all $(w, v) \in S$. The size of a sample S is defined as $|S| = \sum_{(w,v)\in S} (|w| + |v|)$.

Definition 6 ($(\mathbb{T}, \mathbb{R})$-Learning Algorithm)**.** A $(\mathbb{T}, \mathbb{R})$-learning algorithm $\mathcal{A}$ is a program that takes as input a sample S for some $t \in \mathbb{T}$ and returns a representation $r \in \mathbb{R}$.

Here the class $\mathbb{R}$ consists of pairs of Simplex ISL_2 transducers (T_f,T_g). The mapping N from representation to target functions is $N(T_f, T_g) \stackrel{\text{def}}{=} f(T_g) \circ f(T_f)$.

Definition 7 (Strong characteristic sample)**.** For a $(\mathbb{T}, \mathbb{R})$-learning algorithm $\mathcal{A}$, a sample CS is a *strong characteristic sample* of a representation $r \in \mathbb{R}$ if for all samples S for $\mathcal{L}(r)$ such that $CS \subseteq S$, $\mathcal{A}$ returns r.

Definition 8 (Strong identification in polynomial time and data (de la Higuera, 2010))**.** A class $\mathbb{T}$ of functions is *strongly identifiable in polynomial time and data* if there exists a $(\mathbb{T}, \mathbb{R})$-learning algorithm $\mathcal{A}$ and two polynomials p and q such that, for any characteristic sample S of size m for $r \in \mathbb{R}$, $\mathcal{A}$ returns a hypothesis $r \in \mathbb{R}$ in $O(p(m))$ time, and for each representation $r \in \mathbb{R}$ of size k, there exists a strong characteristic sample of r for $\mathcal{A}$ of size at most $O(q(k))$.

6 Algorithm

In this section we give the Simplex ISL2 Function Interaction Decomposition Algorithm (SI_2FIDA). Since both f and g are subsequential functions, their composition $g \circ f$ is also subsequential. Consequently, $g \circ f$ can be learned in the limit from positive data in cubic time using the Onward Subsequential Transducer Inference Algorithm (OSTIA; (Oncina et al., 1993). The inference procedure therefore begins with the transducer returned by OSTIA, which is constructed from input-output pairs representing f, g, and their composition.

Algorithm 1: SI_2FIDA

Data: A finite function $D : P^* \to \Sigma^*$

$T_h := \text{OSTIA}(D) = \langle Q_h = \{q_1, ..., q_n\} = Q_{h\rtimes},\ q_{0_h},\ \delta_h,\ \omega_h,\ \iota_h\rangle$;

$IS(q) = \{suff_1(\rho \text{ s.t } \delta_h(q, \rho) = q)\}$;

$q_{def} \leftarrow q$ s.t. $\forall\sigma \in \Sigma, \omega(q, \sigma) = \sigma$;

$q_{extra} \leftarrow q$ s.t. $\exists(\sigma, \sigma' \in \Sigma, q' \neq q_{\text{def}} \neq q)$ s.t. $\delta(q_{\text{def}}, \sigma) = q'$ and $\delta(q', \sigma') = q$ and $\neg\exists\sigma$ s.t. $\delta(q_{\text{def}}, \sigma) = q$;

if $|(Q_h - \{q_{extra}\}| = 3$ **then**

  $q_{env_1} \leftarrow q \in Q_h - \{q_{extra}\}$ s.t $q \neq q_{def}$;

  $q_{env_2} \leftarrow q \in Q_h - \{q_{extra}\}$ s.t $q \neq q_{def} \wedge q \neq q_{env_1}$

if $|(Q_h - \{q_{extra}\}| = 2$ **then**

  $q_{env_1} \leftarrow q \in Q_h - \{q_{extra}\}$ s.t $q \neq q_{def}$;

  $q_{env_2} \leftarrow q \in Q_h - \{q_{extra}\}$ s.t $q \neq q_{def}$

$T_f, T_g := \texttt{construct_}T_fT_g(T_h)$

return $\texttt{order_}T_fT_g(T_h, T_f, T_g)$

Algorithm 1 takes as input a finite set of inputoutput pairs. The first step is to construct a transducer that represents the composition of two functions using OSTIA. The algorithm then identifies the state whose outgoing transitions are all identity transitions as the default state, denoted q_{def}. Next, it identifies q_{extra}, the state that is reachable from q_{def} only via a path of length two. If removing q_{extra} leaves three states in the transducer, the two remaining non-default states are denoted q_{env_1} and q_{env_2}, respectively. If removing q_{extra} leaves only two states, then both q_{env_1} and q_{env_2} refer to the unique non-default state in T_h. The

Algorithm 2: construct_$T_f T_g$

Data: $T_h := \text{OSTIA}(D) = \langle Q_h = Q_{h\rtimes},\ q_{0_h},\ \delta_h,\ \omega_h,\ \iota_h \rangle$

1 **if** $q_{0_h} = q_{env1}$ *or* $q_{0_h} = q_{env2}$ **then**
2 $\quad T_f := \langle Q' = \{q'_{def}, q'_{env}\} = Q'_{\rtimes},\ q'_0 = q_{0_h},\ \delta',\ \omega',\ \iota' \rangle$;
3 $\quad T_g := \langle Q'' = \{q''_{def}, q''_{env}\} = Q''_{\rtimes},\ q''_0 = q_{def},\ \delta'',\ \omega'',\ \iota'' \rangle$;
4 **else**
5 $\quad T_f := \langle Q' = \{q'_{def}, q'_{env}\} = Q'_{\rtimes},\ q'_0 = q_{0_h},\ \delta',\ \omega',\ \iota' \rangle$;
6 $\quad T_g := \langle Q'' = \{q''_{def}, q''_{env}\} = Q''_{\rtimes},\ q''_0 = q_{0_h},\ \delta'',\ \omega'',\ \iota'' \rangle$;
7 **for** $\sigma \in \Sigma$ **do**
8 $\quad$ // Establish transitions in T_f for σ ;
9 $\quad$ **if** $\delta_h(q_{def}, \sigma) = q_{env_1}$ **then** $\delta'(q'_{def}, \sigma) := q'_{env}$;
10 $\quad$ **else** $\delta'(q'_{def}, \sigma) := q'_{def}$;
11 $\quad$ **if** $\delta_h(q_{env_1}, \sigma) = q_{env_1}$ **then** $\delta'(q'_{env}, \sigma) := q'_{env}$;
12 $\quad$ **if** $\delta_h(q_{env_1}, \sigma) = q_{def}$ **then** $\delta'(q'_{env}, \sigma) := q'_{def}$;
13 $\quad$ **if** $\delta_h(q_{env_1}, \sigma) = q_{extra}$ **then**
14 $\quad\quad$ **if** $\sigma = IS(q_{env_1})$ **then** $\delta'(q'_{env}, \sigma) := q'_{env}$;
15 $\quad\quad$ **else** $\delta'(q'_{env}, \sigma) := q'_{def}$;

Algorithm 2: construct_$T_f T_g$ (continued)

16 **for** $\sigma \in \Sigma$ **do**
17 $\quad$ // Establish transitions in T_g for σ ;
18 $\quad$ **if** $\delta_h(q_{def}, \sigma) = q_{env_2}$ **then** $\delta''(q''_{def}, \sigma) := q''_{env}$;
19 $\quad$ **else** $\delta'(q''_{def}, \sigma) := q''_{def}$;
20 $\quad$ **if** $\delta_h(q_{env_2}, \sigma) = q_{env_2}$ **then** $\delta''(q''_{env}, \sigma) := q''_{env}$;
21 $\quad$ **if** $\delta''(q''_{env}, \sigma) := q''_{def}$ **then** $\delta''(q''_{env}, \sigma) := q''_{def}$;
22 $\quad$ **if** $\delta_h(q_{env_2}, \sigma) = q_{extra}$ **then**
23 $\quad\quad$ **if** $\sigma = IS(q_{env_2})$ **then** $\delta''(q''_{env}, \sigma) := q''_{env}$;
24 $\quad\quad$ **else** $\delta''(q''_{env}, \sigma) := q''_{def}$;
25 $change' = (\tau', s')$ s.t. $\omega_h(q_{env_1}, \tau') = s'$ and $\tau' \neq s'$;
26 $change'' = (\tau'', s'')$ s.t. $\omega_h(q_{env_2}, \tau'') = s''$ and $\tau'' \neq s''$;
27 **for** $\sigma \in \Sigma$ **do**
28 $\quad$ **for** $q \in Q'$ **do** $\omega'(q', \sigma) = \sigma$; $\omega'(q'_{env}, \tau') = s'$
29 $\quad$ **for** $q \in Q''$ **do** $\omega''(q'', \sigma) = \sigma$;
30 $\iota'(q'_{def}) = \iota'(q'_{env}) = \iota''(q''_{def}) = \iota''(q''_{env}) = \lambda$
31 **return** T_f, T_g

next step is to decompose this composed transducer into the two underlying component transducers and determine their relative ordering.

Algorithm 2 shows how two transducers, T_f and T_g, are constructed from the composed transducer T_h. The algorithm takes the OSTIA output as input and initializes two simplex ISL_2 transducers in Lines 1-7. Under the simplex ISL_2 assumption, each transducer has a canonical structure consisting of exactly two states: an environment state and a default state. q_{env1} and q_{env2} map to the environment state of each individual machine. If the initial state of the composed machine corresponds to one of the environment states, then the default state in the composed machine maps to the initial state of one individual machine. Otherwise, the initial state of the composed machine functions as the default state for both individual machines.

Lines 8–16 compute the transition function for T_f. Under the simplex ISL_2 assumption, the individual transducer contains two types of transitions: transitions exiting the default state and transitions exiting the environment state. All such transitions in the individual machine can be recovered from the transition behavior of the composed machine. For transitions exiting the default state, if reading σ in T_h causes a transition from the q_{def} to q_{env1}, then the corresponding transition in T_f also goes from the q'_{def} to q'_{env}. Otherwise, the transitions exiting the default state in T_f remain in the default state. Transitions exiting the environment state have two possible outcomes as well: they either loop on the environment state or transition to the default state. If reading σ in the composed machine leads to a transition that loops on q_{env1}, then the corresponding transition in T_f also loops on q'_{env1}. If the composed machine transitions from q_{env1} to q_{def}, then the corresponding transition in T_f also goes from q'_{env} to q'_{def}. Finally, if the composed machine transitions from q_{env1} to q_{extra}, then the transition in T_f loops on q'_{env} if σ matches the input suffix associated with

q_{env1}. All other transitions from q'_{env} transition to q'_{def}. Lines 17-25 then compute the transition function for T_g in the same way.

Lines 26–27 identify the sole non-identity transition in each machine. Under the simplex ISL_2 assumption, non-identity outputs occur only on transitions exiting the environment state. Lines 28–32 then construct the output functions for the two individual machines. By definition, aside from this single non-identity output, all other outputs are identity mappings.

Algorithm 3 determines the ordering once the two individual machines have been derived. Lines 1–6 identify the environment, trigger, and output associated with each machine. Lines 7–12 detect a potential feeding interaction, which is when the environment of one function corresponds to the output of the other. In these cases, if the transition from the environment state goes to the default state, the environment-affecting function is ordered second; otherwise, it is ordered first. Lines 13–15 detect a potential bleeding interaction, where the environment of one function corresponds to the input of the other. In these cases, if the transition from the environment state goes to the other environment state, the environment-affecting function is ordered second; otherwise, it is ordered first.

Algorithm 3: order_T_fT_g

Data: T_h, T_f, T_g

1 $env' \leftarrow$ the unique $e_1 \in \text{IS}_1(q'_{env})$
2 $env'' \leftarrow$ the unique $e_2 \in \text{IS}_1(q''_{env})$
3 $\tau' \leftarrow$ the unique symbol s.t. $\omega'(q'_{env}, \tau') = w'_\tau$, for $w'_\tau \neq \tau'$
4 $s' \leftarrow \text{suff}_1(env'w'_\tau)$
5 $\tau'' \leftarrow$ the unique symbol s.t. $\omega''(q''_{env}, \tau'') = w''_\tau$, for $w''_\tau \neq \tau''$
6 $s'' \leftarrow \text{suff}_1(env''w''_\tau)$
7 **if** $env'' = s'$ **then**
8 **if** $\delta_h(q_{env_1}, \tau') = q_{def}$ **then return** (T_g, T_f)
9 **else return** (T_f, T_g)
10 **if** $env' = s''$ **then**
11 **if** $\delta_h(q_{env_2}, \tau'') = q_{def}$ **then return** (T_f, T_g)
12 **else return** (T_g, T_f)
13 **if** $env'' = \tau'$ **then**
14 **if** $\delta_h(q_{env_1}, \tau') = q_{env_2}$ **then return** (T_g, T_f)
15 **else return** (T_f, T_g)
16 **if** $env' = \tau''$ **then**
17 **if** $\delta_h(q_{env_2}, \tau') = q_{env_1}$ **then return** (T_f, T_g)
18 **else return** (T_g, T_f)
19 **return** (T_f, T_g)

7 Correctness and Complexity

In the following, assume T_f and T_g are the canonical SFSTs for two simplex ISL_2 functions f and g, respectively. Clearly, then, (T_f, T_g) is a unique canonical grammar for $g \circ f$. The below shows how the above algorithm identifies (T_f, T_g) from a sample of $g \circ f$.

The CS for OSTIA for a transducer T is essentially any set of strings that 1) reach each state; 2) exercise each transition; and 3) provide enough information to correctly calculate the longest common prefix of the output on each transition. We refer the reader to Oncina et al. (1993) for details. What is important here is that any CS for OSTIA for T returns a canonical SFST for T.

Lemma 4. The CS for OSTIA for $T_{g\circ f}$ is a CS for (T_f, T_g).

Proof. The CS for OSTIA for $T_{g\circ f}$ generates the canonical transducer for $T_{g\circ f}$ (Oncina et al., 1993). The correctness of the present algorithm then follows from Lemmas 2 and 3. Algorithm 1 identifies the states of two machines T' and T'' in $T_{g\circ f}$ exactly based on the structure of $T_{g\circ f}$ identified in Lemma 2 and Algorithm 2 builds the states and transitions of T' and T'' accordingly. Algorithm 3 then establishes which of T' and T'' is T_f and which is T_g based on the information about environments and targets available in $T_{g\circ f}$ as established in (1) and (2) of Lemma 3. □

Lemma 5. The size of the CS for SI_2FIDA is linear in the size of the grammar.

Proof. The size of the CS for OSTIA for any T is linear in the size of T (Oncina et al., 1993). □

Lemma 6. SI_2FIDA runs in time cubic with respect to the input data.

Proof. OSTIA runs in cubic time (Oncina et al., 1993). The `for` loops Algorithm 3 loop over the symbols in Σ, thus the steps in Algorithms 1 through 3 are constant with respect to the data. □

Theorem 1. SI_2FIDA strongly identifies T_f T_g from the limit in polynomial time and data.

Proof. From Lemmas 4 through 6. □

8 Discussion and Conclusion

Given that process interaction is pervasive in natural language phonology, a child learner must be able to infer not only individual processes but also their interaction. The present work establishes an initial inference procedure for recovering two interacting simplex ISL_2 functions from their composition. This thus takes steps towards solving the heretofore unsolved problem of learning interacting processes from data.

The problem of learning interacting processes is particularly important because process interaction is pervasive in natural language phonology. Chinese tone sandhi systems, for example, exhibit abundant interactions between tonal processes. In the Changting dialect, there are five lexical toneshigh, mid, low, rising, and fallingand 15 of the 25 possible disyllabic tone combinations undergo tone sandhi. Because the tonal inventory is limited, the application of one process frequently affects the environment required by another, giving rise to abundant interactions. Changting exhibits both transparent and opaque interactions. Furthermore, each individual tone sandhi process can be characterized as a simplex ISL_2 ($sISL_2$) function, which is the type of interaction investigated in this study.

The present algorithm exploits the canonical two-state structure of simplex ISL_2 transducers and the constrained form of their composition. A natural open question is thus how to generalize this result. First, the inference procedure should be generalized beyond $k = 2$.

A second direction involves extending the analysis to any arbitrary number of functions. In natural language phonology, the observable mapping may reflect the serial application of multiple, potentially interacting processes. Although subsequential functions are closed under composition, the corresponding decomposition problem is likely to become increasingly complex as the number of component functions grows. However, the current work shows that a strong definition of what it means to be an 'individual process' gives the learner sufficient structure to disentangle function interactions.

Acknowledgements

The authors thank the helpful suggestions of three anonymous reviewers. Both authors were supported in this work by NSF grant #2416184.

References

Paul Ariste. 1968. *A Grammar of the Votic Language*. Bloomington: Indiana University Press.

Eric Baković. 2007. A revised typology of opaque generalisations. *Phonology*, 24(2):217259.

Jane Chandlee. 2014. *Strictly Local Phonological Processes*. Ph.D. thesis, University of Delaware.

Jane Chandlee, Rémi Eyraund, and Jeffery Heinz. 2015. Output strictly local functions. *Proceedings of the 14th Meeting on the Mathematics of Language*.

Colin de la Higuera. 2010. *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press.

Wenyue Hua and Adam Jardine. 2021. Learning input strictly local functions from their composition. In *ICGI 15*, volume 153, pages 47–65. PMLR.

Adam Jardine, Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2014. Very efficient learning of structured classes of subsequential functions from positive data. In *Proceedings of the 12th International Conference on Grammatical Inference (ICGI 2014)*, JMLR Workshop Proceedings, pages 94–108.

Paul Kiparsky. 1973. Abstractness, opacity, and global rules. In Osamu Fujimura, editor, *Three dimensions in linguistic theory*, pages 57–86. Tokyo: TEC.

M. Lothaire. 2005. *Applied Combinatorics on Words*. Cambridge University Press.

David Odden. 2005. *Introducing Phonology*. Cambridge University Press.

José Oncina, Pedro García, and Enrique Vidal. 1993. Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Jacques Sakarovitch. 2009. *Elements of Automata Theory*. Cambridge University Press.

Marcel Paul Schützenberger. 1977. Sur une variante des fonctions séquentielles. *Theoretical Computer Science*, 4:47–57.

Do I know what I want to say? Modeling meaning uncertainty in RSA

Anzi Wang[*1,2], **Carolyn Jane Anderson**[3], **Grusha Prasad**[*2]

[1]Linguistics, Northwestern University
[2]Computer Science, Colgate University
[3]Computer Science, Wellesley College

Correspondence: anziw@u.northwestern.edu, carolyn.anderson@wellesley.edu, gprasad@colgate.edu

Abstract

Models using the Rational Speech Act (RSA) framework typically assume that speakers are certain about the meaning being communicated. In this work we note that there are contexts in which this assumption need not hold, and propose a method (um-RSA) to incorporate this meaning uncertainty within the RSA framework. As a case study, we explore two sources of meaning uncertainty: Counting-Uncertainty (from numerical cognition) and Discounting-Uncertainty (from behavioral economics). We generate predictions from these two hypotheses and test these predictions with two human experiments. The results show that um-RSA can account for differences in uncertainty expression usage that the standard RSA framework cannot account for, thus demonstrating the usefulness of modeling meaning uncertainty.

1 Introduction

The Rational Speech Act (RSA) framework (Frank and Goodman, 2012) views communication as recursive reasoning, where speakers and listeners reason about each others' interpretation of utterances. Prior work has used this recursive reasoning process to effectively model the uncertainty that listeners maintain about speakers and vice versa (Goodman and Stuhlmüller, 2013). Prior work has also modeled the uncertainty that listeners might maintain about properties of the world that is being discussed (Lassiter and Goodman, 2013; Degen et al., 2015; Herbstritt and Franke, 2019) or the question under discussion (Kao et al., 2014a; Kao and Goodman, 2015; Kao et al., 2014b). See Degen (2023) for a review. Despite the differences in the phenomena being modeled, this body of work consistently makes the following assumption: the speaker at the first level of recursion is certain about the target of communication (*target meaning*).

However, this assumption needs not always hold, especially in experiments where participants are presented with images that they are asked to describe. In these experiments, the target meaning for a given trial is often assumed to be specified by the target image presented in that trial, and is thus consistent across speakers (i.e., participants). For example, consider an experiment where participants are presented with images of gumball machines and asked to asked to communicate the probability of the machine dispensing orange gumballs. In a trial where the target image has 18 orange gumballs out of 30, the target meaning is assumed to be $P(red) = 60\%$ (cf. Schuster and Degen 2020).

Yet, uncertainty can emerge in this target meaning based on how the participant *perceives* the target image. Since humans often use an approximate number system (e.g., Gilmore et al. 2011), in a trial with 18 orange gumballs, a participant might approximate the number of red gumballs (e.g., between 15-20), instead of counting to determine the exact number. Such an approximation could result in the participant maintaining uncertainty over the target meaning (e.g., $50\% < P(orange) < 67\%$).

Similarly, uncertainty can also emerge based on how a participant *interprets* the target image. Since people tend to discount low-probability events with high-impact consequences (Sundh, 2024), when presented with such potentially high-impact scenarios, such as election predictions, participants might discount these predictions and assign some mass to lower probabilities. Such discounting, could result in participant maintaining uncertainty over the target meaning (e.g., $P(\text{X winning}) \leq 30\%$).

Drawing on these scenarios, we introduce an extension to the RSA framework that models speakers' uncertainty over the target of communication. We call this extension the *Uncertain Meaning-RSA* or um-RSA.[1] Our approach can be used to model

[*] Equal contribution to project conceptualization and development. AW led the implementation of the RSA models and execution of the human and model experiments. GP led the statistical analyses and writing.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 313–328
July 3-4, 2026 ©2026 Association for Computational Linguistics

any hypothesis about uncertainty over meaning.

We use um-RSA to generate predictions for the two (not mutually exclusive) hypotheses about uncertainty described above: First, meaning uncertainty can emerge from imprecision in speakers' perception of exact numbers in images (Counting-Uncertainty). Second, meaning uncertainty can emerge from speakers discounting lower probability events (Discounting-Uncertainty). Specifically, we generate predictions for three types of utterances: *bare not*, *might*, and *probably*. The predicted patterns indicate that compared to Counting-Uncertainty, Discounting-Uncertainty results in increased preference for *bare not* and *probably* (i.e., utterances that express more confidence about the outcome), a difference not predicted by the standard RSA framework.

We test these predictions in two within-subject experiments.[2] In the first experiment participants reasoned about images with gumballs and election predictions. We hypothesized participants would discount probabilities in the higher-impact election condition, but not lower-impact gumball condition (cf. Sundh 2024). Consistent with this, we found that participants produced the more confident *probably* and *bare not* utterances more frequently in the election compared to the gumball condition. To further investigate the source of this difference, in a follow-up experiment, we replaced images with text descriptions, thus eliminating uncertainty due to numerical approximations. We found the same qualitative pattern of results, lending support to the hypothesis that the difference between the conditions was likely driven by participants discounting lower probabilities in election predictions.

Thus, our work makes the following contributions. First, we note that the standard RSA framework assumes that speakers are always certain about the target of communication, but this assumption needs not always hold. Next, we propose an extension, um-RSA, that can be used to implement *any* hypothesis about speakers' uncertainty over target meaning. Finally, we demonstrate the usefulness of this approach by generating and testing predictions from two hypotheses.

[1]The 'um' is not just acronym but also captures the sound speakers might make when they are uncertain!

[2]Our model implementation, data, and statistical analyses can be found here: https://github.com/anziw/uncertainty-exp

2 Background

2.1 Rational Speech Act (RSA) Framework

RSA frames conversation as a process in which speakers choose an utterance u given a meaning m they wish to convey, and listeners infer a meaning m given the utterance u they have heard. As a rational framework, RSA assumes that the speaker selects an utterance that maximizes the probability that the listener correctly infers the intended meaning. Accordingly, the framework is built around two core sets: a set of possible meanings a speaker may intend to communicate (M) and a set of possible utterances that can be used to communicate those meanings (U). In many experimental settings, both communicative goals and available utterances are constrained by the design of the task; as a result, RSA models in these contexts typically assume that M and U are finite and relatively small.

Simulating a speaker's probability distribution over utterances requires specifying four components. First, a literal meaning mapping from U to M: for every m_i and utterance u_j, $[\![u_j]\!](m_i)$ is True if the speaker *could* veridically use u_i to convey m_i, and False otherwise. Second, a prior $P(m)$ which captures a listener's subjective probabilities for each meaning in M being the target of communication in the current context. Third, an utterance cost $C(u)$ which specifies the cost of each utterance in U. Finally, a rationality parameter λ which governs how likely the speaker is to choose the utterance that maximizes utility.

As discussed earlier, this framework is also inherently recursive: the speaker reasons about the listener, who in turn reasons about the speaker, who reasons again about the listener, and so on. In this work, we focus on only the first level of recursion, in which a pragmatic speaker S_1 reasons about a literal listener L_0, but the mechanism we propose can be as well extended to higher levels of recursion (see Section 7.1). Simulating S_1 involves two steps. First, the L_0 is simulated by multiplying the literal meaning $[\![u]\!](m)$ by the prior $P(m)$, and normalizing over all meanings to get the probability distribution over meanings conditioned on utterances ($P(m \mid u)$). Second, S_1 is simulated by incorporating the utterance cost $C(u)$ and rationality parameter λ into the L_0 conditional distribution, and then normalizing over all utterances to obtain the probability distribution over utterances conditioned on meanings ($P(u \mid m)$). The specific formulae for these steps are outlined in Table 1.

2.2 Modeling uncertainty with RSA

Prior work studying uncertainty expressions, such as *might* and *probably*, found that there is considerable inter-speaker variability in uncertainty expression usage (Wallsten et al., 1986; Yildirim et al., 2016), and that listeners adapt to these speaker-specific uses (Schuster and Degen, 2020). Schuster and Degen used the RSA framework to model this adaptation by assuming that speakers use different thresholds to determine the literal meaning of these utterances. For example, a confident speaker might consider *probably* to be literally true when describing any event with a probability greater than 60%, whereas a more cautious speaker, adopting a higher threshold, may consider *probably* to be literally true only for events with greater than 80% probability. Listener adaptation to this was modeled as listeners learning these speaker-specific thresholds.

Prior work has also demonstrated the limitations of the RSA framework and proposed extensions. For example, Degen et al. (2015) show that RSA models overestimate the influence of prior world knowledge. Consider the following sentence "Some of the marbles sank", which elicits the scalar implicature that some, but not all of the marbles sank. Standard RSA does not predict this implicature because the prior for all of the marbles sinking is very high given world knowledge. Degen et al. propose an extension in which listeners additionally reason about whether the world being described is normal or "wonky"; if it is a "wonky" world, then they back off to a uniform prior.

As another example, Kao et al. (2014b) show that standard RSA cannot model listeners' imprecise interpretations of non-literal utterances. For example, consider the following utterances:

(1) It took me 62 minutes to get here
(2) It took me 60 minutes to get here
(3) It took me 2 million years to get here

Kao et al. note that listeners only interpret the first utterance literally. Listeners might interpret (2) as the speaker took around, but not exactly, 60 minutes, and (3) as it took the speaker a very long time, but not literally 2 million years. Standard RSA prioritizes utterances that are literally true, and thus cannot model this phenomenon. To overcome this, Kao et al. propose an extension in which speakers can convey affect through their utterances, and can choose to convey the affect or not depending on their goals. In addition to inferring the meaning, listeners also infer speakers' goals and affects.

Despite the different types of variability and uncertainty modeled in prior work and the extensions proposed to the standard RSA framework, there is one central assumption that this body of work makes: the speaker at the first level of recursion is certain about the meaning they intend to communicate. In the following section we discuss why this assumption need not always hold.

2.3 Motivating uncertainty in meaning

There are at least two reasons why a speaker might be uncertain about the meaning being communicated. First, the speaker may be uncertain about the **communicative goal** itself, such as when attempting to express an idea or emotional state that has not yet been fully fleshed out. Second, the speaker may be uncertain about **the information used to form that communicative goal**, such as a speaker who is sure that their goal is to communicate the likelihood of rain based on a weather forecast, but is unsure about how trustworthy the prediction is. In this work, we focus on the latter type of uncertainty. Specifically, we examine two forms of uncertainty that could arise in experimental settings in which participants' intended meanings in any given trial are determined by how they perceive or interpret probabilistic information they are presented with.

Counting-Uncertainty This type of uncertainty could arise when participants determine the intended meaning by estimating counts of objects (e.g.,gumballs) in target images. Prior work on people's numerical representations has demonstrated that for sufficiently large numbers, instead of counting objects individually, people rely on an Approximate Number System (ANS) (Gilmore et al., 2011). Since this system is *approximate*, participants might maintain uncertainty about the exact number of objects in the target image, and thus the exact probability they intend to communicate.

Discounting-Uncertainty This type of uncertainty could arise when participants determine the intended meaning by estimating probabilities of high-stake events (e.g., election outcomes) by reasoning about probabilistic information, such as election predictions. Previous work in behavioral economics and decision making has shown that people have difficulty reasoning about low probability events (Kahneman and Tversky, 1979), and tend to underweight or discount low probabilities for high-impact events (Sundh, 2024). Due to such discounting, participants might maintain uncertainty

Phase	Step	Notation/Equation	Implementation details
Init. world	Define set of observed meanings	M^O	$M^O \in \{m^O_{10\%}, m^O_{20\%} \dots m^O_{90\%}\}$
	Define the set of utterances	U	$U \in \{u_{probably}, u_{might}, u_{not}\}$
	Define set of intended meanings	M^I	$M^I \in \{m^I_{10\%}, m^I_{20\%} \dots m^I_{90\%}\}$
	Initialize weight matrix	$W^{O\to I} \in \mathbb{R}^{\lvert M^O\rvert \times \lvert M^I\rvert}$	$W^{O\to I} \in \mathbb{R}^{9\times 9}$
Init. speaker	Specify literal meaning	$[\![\text{u}]\!](m)$	Sample thresholds $t_{not}, t_{might}, t_{prob}$ $[\![u_{not}]\!](m)$ = 1 if $m \le t_{not}$, else 0 $[\![u_{might}]\!](m)$ = 1 if $m \ge t_{might}$, else 0 $[\![u_{prob}]\!](m)$ = 1 if $m \ge t_{prob}$, else 0
	Specify prior for each meaning	$P(m^O)$	Uniform prior. $P(m^O_i) = \frac{1}{9}$ for all states.
	Specify utterance cost	$C(u)$	Constant cost. $C(u_i) = 3.03$
	Specify rationality parameter	λ	Constant across speakers. $\lambda = 2.211$
	Set weights of $W^{O\to I}$	$W^{O\to I} = W^h$	3 hypotheses (h) to set weights (see § 4.1)
Simulate Committed Speaker	Literal Listener	$L^O_0(m^O \mid u) = norm([\![\text{u}]\!](\text{m}^O) \cdot P(m^O))$	*norm* normalizes across states
	Move to intended meaning	$L^I_0(m^I \mid u) = (norm(W^{O\to I}) \cdot (L^O_0)^\top)^\top$	$\top$ is transpose; *norm* across states
	Pragmatic reasoning	$S^I_1(u \mid m^I) = norm(exp(\lambda(logL^I_0(m^I \mid u) - C(u))))$	*norm* normalizes across utterances
Simulate Uncommitted Speaker	Literal Listener	$L^O_0(m^O \mid u) = norm([\![\text{u}]\!](\text{m}^O) \cdot P(m^O))$	*norm* normalizes across states
	Pragmatic reasoning	$S^O_1(u \mid m^O) = norm(exp(\lambda(logL^O_0(m^O \mid u) - C(u))))$	*norm* normalizes across utterances
	Move to intended meaning	$S^I_1(m^I \mid u) = (norm(W^{O\to I}) \cdot (S^O_1)^\top)^\top$	$\top$ is transpose; *norm* across utterances

Table 1: Formulae for the standard RSA framework (black) and um-RSA (red).

about the exact probability of a high-stakes event occurring given some target prediction, and thus the exact probability they intend to communicate.

Between-participant differences in intended meaning In addition to the *within* participant uncertainty described above, the same observed meaning could result in different intended meanings *between* participants given that prior work has found individual differences in how precisely participants approximate counts (Halberda et al., 2008) or discount low probability events (Khaw et al., 2021)

3 The Uncertain-Meaning RSA: um-RSA

We introduce meaning uncertainty in our model um-RSA by differentiating between two types of meaning: observed meaning (m^O) and intended meaning (m^I). For any given trial t, m^O_t refers to the meaning specified by the information presented in t. Standard RSA models assume that the target of communication for t is m^O_t. In contrast, um-RSA assumes that the target of communication is m^I_t, i.e., the meaning that participants *derive* from the information presented to them in t. Meaning uncertainty emerges from the derivation process.

Specifically, um-RSA defines separate sets for observed and intended meanings, M^O and M^I respectively, and uses a weight matrix $W^{O\to I} \in \mathbb{R}^{\lvert M^O\rvert \times \lvert M^I\rvert}$ to convert probabilities over observed meanings to probabilities over intended meanings. In this work we assume that $\lvert M^O\rvert$ and $\lvert M^I\rvert$ are the same, but this assumption is not necessary.

3.1 Specifying the weights

The weights of $W^{O\to I}$ can be specified in different ways depending on the modeling goal. When the goal is hypothesis testing, the weights can be manually specified to encode specific hypotheses about meaning uncertainty. In this work, we construct three versions of $W^{O\to I}$ to implement our two hypotheses, Counting-Uncertainty and Discounting-Uncertainty, and our baseline No-Uncertainty (see § 4.1). In contrast, when the goal is to infer or characterize meaning uncertainty from data, the weights can be learned using standard optimization techniques. Different approaches for learning these weights are briefly discussed in § 7.1.

3.2 Applying the weights

There are two stages at which $W^{O\to I}$ can be applied in the standard RSA framework, which results in two models of pragmatic speakers that incorporate meaning uncertainty in slightly different ways. First, *a committed speaker model*, where the speaker has uncertainty over the intended meaning initially, but settles on a unique intended meaning before applying pragmatic reasoning to select an utterance. Second, *an uncommitted speaker model*, where the speaker does not commit to a specific intended meaning, but rather marginalizes over possible intended meanings after pragmatic reasoning. We describe these two models below in more detail, and include specific formulae in Table 1.

Committed speaker model Like in the standard RSA framework, the model starts with the literal

listener estimating $L_0^O(m^O \mid u)$, a probability distribution over *observed meanings* conditioned on available utterances. The listener then moves into the intended meaning space by multiplying $L_0^O(m^O \mid u)$ by $W^{O \to I}$, yielding $L_0^I(m^I \mid u)$. Finally, the pragmatic speaker S_1 incorporates the utterance cost and rationality parameter to estimate $S_1^I(u \mid m^I)$, a probability distribution over utterances conditioned on intended meanings. This model maintains the standard RSA assumption that the communicative target is fixed during pragmatic reasoning. Because the model transitions to the intended meaning space before estimating S_1, utterances are evaluated conditioned on intended meanings, effectively committing to an intended communicative target during utterance selection.

Uncommitted speaker model Like in the committed speaker model, this model starts with $L_0^O(m^O \mid u)$. Then, instead of moving into the intended space, this model incorporates the cost and rationality parameter to estimate $S_1^O(u \mid m^O)$, a probability distribution over utterances conditioned on the set of *observed meanings*. Finally, the pragmatic speaker moves into the intended space by multiplying $S_1^O(u \mid m^O)$ by $W^{O \to I}$ to estimate $S_1^I(u \mid m^I)$, a probability distribution over utterances conditioned on *intended meanings*. Since the transition to intended meaning space occurs only after pragmatic reasoning, utterances are optimized without reference to a fixed intended meaning, effectively maintaining uncertainty over communicative targets during pragmatic reasoning.

4 Model experiment

As a case study, we model an experimental setting in which speakers describe 9 possible probabilistic events evenly spaced between 10% to 90% using three possible utterances: not, might, and probably. Thus we operate over the following sets:

- $M^O = M^I = \{m_{10\%}, m_{20\%} \ldots m_{90\%}\}$
- $U = \{u_{not}, u_{might}, u_{probably}\}$

We generate predictions from um-RSA using 3 hypotheses about the type of meaning uncertainty.

4.1 Hypotheses → Weights

No-Uncertainty This hypothesis assumes there is no meaning uncertainty in speakers. Thus observed meaning is identical to the intended meaning. The weight matrix encoding this hypothesis is an identity matrix where the values of the principal diagonal are 1, and all other values 0.

Counting-Uncertainty This hypothesis assumes that speakers are uncertain about intended meaning because their counts of objects in target images are imprecise. Specifically, we assume that when the observed meaning is $P^O(event) = x\%$ the weight matrix has non-zero values for $P^I(event) > x - 10\%$ and $P^I(event) < x + 10\%$, with a value of 1 for $P^O(_{\text{event}}) = x\%$ (i.e., the principal diagonal) The non-zero values are determined by a decay parameter d such that intended probabilities further away from x are assigned lower values.

Discounting-Uncertainty This hypothesis assumes that speakers are uncertain about intended meaning because they discount probabilities in target predictions. Specifically, we assume that when the observed meaning is $P^O(event) = x\%$, the weight matrix has non-zero values for $P^I(event) \leq x$ when $x < 50\%$, and $P^I(event) \geq x$ when $x > 50\%$, with a value of 1 when $P^I(event) = x$. As with Counting-Uncertainty an exponential decay d assigns lower values to probabilities further away from x.

4.2 Generating predictions

We use um-RSA to simulate individual speakers (analogous to participants in a human experiment), and compute the overall predictions by aggregating across all of these speakers. Each speaker is determined by two factors: the thresholds that determine the literal meaning for the three utterances and the decay parameter that determines the non-zero values in the Counting-Uncertainty and Discounting-Uncertainty weight matrices. We sample the thresholds using the following distributions from Schuster and Degen (2020): $t_{not} \sim Beta(0.407, 1.219)$, $t_{might} \sim Beta(0.928, 3.086)$, $t_{probably} \sim Beta(2.552, 1.771)$. Additionally, we enforce the following constraint to capture typical assumptions about the restrictiveness of these expressions: $t_{not} \leq t_{might} \leq t_{probably}$. We sample the decay parameter from $Normal(0, 1)$. Given the thresholds and decay parameter for a participant, for each hypothesis, we follow the sequence of steps outlined in Table 1 to generate the probability of the 3 utterances for all 9 meanings.

4.3 Results

In Figure 1, we present the aggregate predictions for simulated speakers under the three hypotheses (Counting-Uncertainty, Discounting-Uncertainty, and No-Uncertainty) for the two speaker models

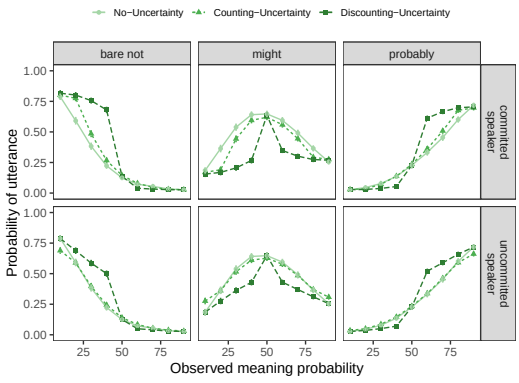


Figure 1: Model predictions under different hypotheses with meaning uncertainty added after the Literal Listener step (top) and after the Pragmatic Speaker step 1. Error bars are two SEs.

(committed speaker and uncommitted speaker). Overall, across the three hypotheses and both speaker models, the simulated speakers displayed patterns consistent with uncertainty expression displayed in Schuster and Degen (2020): $P(not)$ was highest for low probability events, $P(probably)$ was highest for high probability events, and $P(might)$ displayed an inverse u-shape with the highest values for probability events around 50%.

Compared to the No-Uncertainty baseline, the Discounting-Uncertainty condition in both speaker models assigned higher probabilities to utterances which express more confidence towards the event outcome. For events with less than 50% observed probability, Discounting-Uncertainty speakers assigned higher probability to *not* than baseline speakers. Similarly, for events with greater than 50% observed probability, Discounting-Uncertainty speakers assigned higher probabilities to *probably*. The differences between the Discounting-Uncertainty and baseline speakers were greater in the committed speaker model.

In contrast, Counting-Uncertainty and No-Uncertainty condition were considerably more similar to each other, with the conditions being nearly identical in the uncommitted speaker model. While the Counting-Uncertainty condition in the committed speaker model did assign higher higher probabilities to *not* and *probably* for observed probabilities less than and greater than 50% respectively, the qualitative patterns were different from the Discounting-Uncertainty condition.

Taken together, these results predict that when comparing stimuli with the same observed probabilities, stimuli that elicit Discounting-Uncertainty are likely to result in a higher proportion of utterances that express confidence in the outcome (e.g., *not* and *probably*) compared to stimuli that elicit Counting-Uncertainty. This difference is not predicted in the standard RSA framework because it assumes that utterance selection is conditioned on *observed* and not *intended* meanings.

5 Human Experiment 1

The goal of this experiment was to test the predictions from um-RSA. To this extent, we asked participants to reason about gumball machines (cf. Schuster and Degen 2020) and election predictions for an unknown country presented as bars.

Since participants are likely to consider election outcomes to have high impact, and gumball machines to have low impact, we hypothesized that the election, but not the gumball, condition would pattern with Discounting-Uncertainty. Similarly, since probabilistic reasoning about the gumball machine, but not election prediction, requires participants to estimate counts of objects, we hypothesized that the gumball condition, but not the election condition, would pattern with Counting-Uncertainty.

Crucially, since we adopt a within-participant design, for each participant we elicit utterances in the gumball and election conditions for the same observed probabilities. Thus, if participants condition their utterances on *observed probabilities*, like standard RSA assumes, we would not expect to find a difference between these conditions. In contrast, if participants condition their utterances on the derived *intended probabilities*, we would expect the election condition to elicit a higher proportion of *not* and *probably* than the gumball condition.

5.1 Methods

Participants We recruited 120 participants with a US-based IP address, via Prolific. The median completion time was 29 minutes and 47 seconds, and participants received $5.96.

Stimuli and Procedure The experiment was hosted on PCIbex (Zehr and Schwarz, 2018). We used a within-subject blocked design: in the first block participants reasoned about gumball machines, and in the second block about election predictions. Each participant was exposed to 36 trial types: 2 event types (gumball or election), 2 outcome types (for gumball condition, machine dispensing purple or orange gumballs; for election condition, party X or party Y winning), and 9

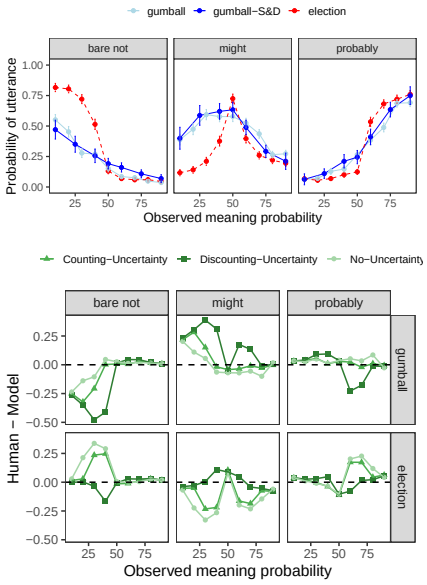


Figure 2: Results from Experiment 1 (top). Average deviation from committed speaker model predictions across 1000 instances (bottom). Error bars are two SEs.

observed probabilities (10% to 90% likelihood of that outcome, in increments of 10%). Participants were presented with each trial type twice, thus resulting in 72 total trials, randomized within each block. Additionally, there were 12 randomly inserted attention checks in each block which asked participants to select the scene they had just seen from two options. At the end of the experiment, participants filled out a demographic survey.

In each trial, participants were presented with a scene, and a person who couldn't see the scene asking a question. In the gumball trials, the scene involved a gumball machine, and the person asked "Will I get a purple/orange gumball?". In the election trials, the scene involved an election prediction, and the person asked "Will party X/Y win the election?". Participants were presented with three possible answers to the questions using *bare not*, *might*, or *probably*. They indicated their relative preference for the utterances with sliders, being required to move at least one slider before proceeding to the next trial. We normalized the slider values into probabilities as a postprocessing step (see § C for a justification). Before each block participants were required to demonstrate that they understood the task, the scene, and how to use the sliders by answering comprehension questions (see § E.2).

Data exclusion We excluded 34 participants because their accuracy on attention checks was below 85% or they reported being non-native speakers.

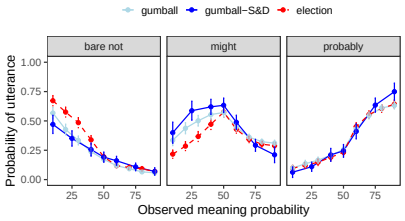


Figure 3: Experiment 2 results with 2*SE error bars.

Statistical analyses Since we were interested in drawing inferences separately for each utterance, instead of using a multinomial logistic regression, we decided to model the probability of each utterance separately with three logistic regression models. Specifically, for each utterance, we fit a mixed effects logistic regression model with the following fixed effects: event type (gumball vs. election), observed probability (50% vs. 10-40%; 50% vs. 60-90%), and the interaction between the two. We included the maximal random effects structure with by-participant random intercept and slopes (see Appendix A).

We decided to model observed probability as a factor with three levels (10-40%, 50%, 60-90%; 50% baseline) instead of a continuous predictor because our experimental design included only 9 observed probabilities, making this predictor not truly continuous. Additionally, for the inferences we want to draw, which are about the overall differences between gumball and election events, the coarse grained distinction of "less than 50%" and "greater than 50%" is sufficient, and thus we did not opt to fit the more complex (and difficult to interpret) mixed effects ordinal logistic regression.

5.2 Results

This experiment uses slightly different stimuli, instructions, and utterance sets from Schuster and Degen (2020) (see § C). Despite these differences, the probabilities of the three utterances in the gumball condition aligned closely (see Figure 2). The slight difference in *bare not* was likely driven by differences in utterance sets (see § C.1). This close replication validates our experimental setup.

Crucially, unlike what standard RSA models predict, there was a significant interaction between event type and observed probability for all three utterances. The effects for *bare not* and *probably* ($P(not \mid 10-40\%) - P(not \mid 50\%)$ and $P(probably \mid 60-90\%) - P(probably \mid 50\%)$) were greater in the election compared to gum-

ball conditions. In contrast, the effects for *might* ($P(might \mid 10-40\%) - P(might \mid 50\%)$ and $P(might \mid 60-90\%) - P(might \mid 50\%)$) was greater in the gumball compared to election condition (see Table 2 for coefficients and p-values).

To compare these empirical results to our um-RSA model predictions, we computed $P_{Human}(u \mid m^O) - P_{Model}(u \mid m^O)$, for each utterance ($u$) and observed meaning ($m^O$) combination, and each of our 3000 model instances (1000 models for each hypothesis). As hypothesized, the Discounting-Uncertainty models better capture the election patterns than the other models (see Figure 2 and § B).

Taken together, these results suggest that speakers are more likely to use expressions that express confidence in the outcome (e.g., *not* and *probably*) when describing election predictions than gumball machines. Our models suggest that this difference is more likely to emerge from the fact that participants *discount* probabilities when reasoning about election predictions. However, since under the committed speaker model, the Counting-Uncertainty condition also resulted in a slight increase in *not* and *probably* compared to the No-Uncertainty condition, we cannot be sure that the difference between gumball and election conditions was not driven by difference degrees of numerical approximation of the observed probability. To test this, we ran a follow up experiment with one crucial difference: we presented participants with text descriptions of the scenes, thus eliminating the need to approximate the observed probability.

6 Human Experiment 2

6.1 Methods

Participants Like with Experiment 1, US-based participants were recruited from Prolific. Each participant received $4.00. The median completion time of the experiment was 27 minutes, 54 seconds.

Stimuli and Procedure We used the same blocked-design and scenes as in Experiment 1, but the images in stimuli were replaced with text descriptions (see § E.3). Initial pilots indicated that these text descriptions significantly increased the duration of the experiment. Therefore, to minimize participant fatigue, we cut down the number of trials by half by asking participants about only one type of outcome, and instead counterbalanced the outcome type between participants (i.e., half the participants were asked questions about orange gumballs and party X, whereas the other half were asked questions about purple gumballs and party Y). The procedure was the same as in Experiment 1.

Data exclusion and Statistical analyses We used the same exclusion criteria and statistical models as in Experiment 1 and excluded 5 participants.

6.2 Results

We see the same qualitative pattern of results as in Experiment 1: for all utterances, we see a significant interaction between observed probability and event type such that *not* is used more frequently in the 10-40%, and *probably* is used more frequently in the 60-90% compared in the election compared to the gumball condition; *might* is used more frequently across the board in the gumball compared to election condition (see Table 3 for coefficients and p-values from our statistical models). Since the text descriptions eliminate Counting-Uncertainty, these results support the inference that the difference between the conditions is likely driven by Discounting-Uncertainty in the election condition.

Note, the preference for *not* and *probably* is smaller in this experiment compared to Experiment 1, especially in the election condition. One possible explanation for this is that discounting of low-probability high-impact events is more likely when the events are experienced rather than described (cf. Sundh 2024), and that our images were more experiential than our text descriptions. This needs to be explored more rigorously in future work.

7 Discussion

In this work, we argue that a common assumption made by most modeling work using the RSA framework — that the speaker is certain about the meaning being communicated — might not always hold: speakers might be uncertain about the meaning (i.e., there is *meaning uncertainty*) either because they are uncertain about the communicative goal itself, or about the information used to form the communicative goal. We propose a generalized approach, um-RSA, where we model meaning uncertainty within the RSA framework by using a weight matrix $W^{O \to I}$ to transform observed meanings to intended meanings. As a case study, we draw on prior work on numerical cognition and behavioral economics to propose two potential sources of meaning uncertainty: Counting-Uncertainty and Discounting-Uncertainty. We use um-RSA to generate predictions from these two hypotheses, and test these predictions with two human experiments.

Our model predictions reveal that for the same observed probabilities, compared to Counting-Uncertainty the preference for utterances that express more confidence about the outcome like *not* and *probably* is higher in Discounting-Uncertainty, a difference not predicted in standard RSA. Consistent with this, our first human experiment revealed that people have a stronger preference for using *not* and *probably* when reasoning about election predictions (hypothesized to elicit Discounting-Uncertainty) than when reasoning about gumball machines (hypothesized to elicit Counting-Uncertainty). Taken together with our modeling results, we tentatively concluded that the difference between the gumball and election conditions was more likely driven by Discounting-Uncertainty than by Counting-Uncertainty. In a follow-up experiment we eliminated Counting-Uncertainty by using text descriptions and found that the difference persisted, thus further supporting our conclusion.

7.1 Future work

Committed vs. Uncommitted speaker models As noted in § 3, there are two different stages at which meaning uncertainty can be incorporated: the committed speaker model and uncommitted speaker model. While these two approaches resulted in qualitatively similar predictions in our experimental setup (see Figure 1), they have different theoretical implications: in the uncommitted model, since the move to intended meaning space occurs *after* pragmatic reasoning, the cost of an utterance can influence the intended meaning. Thus, future work can adjudicate between these two models by experimentally manipulating the cost of utterances.

Other sources of meaning uncertainty Apart from the two sources of meaning uncertainty we considered, there are at least two other categories of meaning uncertainty sources that future work can explore: First, *perceptual uncertainty*, such as from noisy audio or occluded objects in images; Second, *epistemic uncertainty*, such as from confirmation bias or (mis)trust in information sources.

Impact of meaning uncertainty on listeners Since we only focused on the impact of meaning uncertainty on the speaker, this leaves open the question of what impact (if any) this uncertainty might have on listeners. Incorporating meaning uncertainty at the listener level requires first specifying a $W^{O \to I}$ for the listener. There are at least two approaches to specifying this: First, assume that each listener uses their own $W^{O \to I}$, resulting in a potential mismatch between the speaker and listener. Second, assume that listeners adapt to a speaker's $W^{O \to I}$, like they would adapt to other speaker specific preferences. To adjudicate between these approaches, future work can run simulations with and without a mismatch to identify circumstances under which these two approaches generate different predictions, and then test these predictions with human experiments.

Learning the weights of $\mathbf{M^{O \to I}}$ If there is evidence that listeners do adapt to speaker specific $W^{O \to I}$, future work can then explore *what* listeners are adapting to, and *how*. One possibility is to assume a maximally flexible learning mechanism that uses standard optimization techniques to learn each of the weights in the matrix. However, since the number of parameters are directly proportional to the number of utterances and meanings being considered, this approach could prove to be challenging as the spaces of meanings and utterances increase. An alternative is to adopt a more constrained hierarchical approach where a listener first selects the *type* of meaning uncertainty from a small set of hypotheses (e.g., Counting-Uncertainty or Discounting-Uncertainty), and then uses a small set of parameters (e.g., just the decay parameter in our case) to create the $W^{O \to I}$. The learner could then jointly optimize over the hypothesis space, and the parameters for each hypothesis.

7.2 Conclusion

We noted that there are contexts in which speakers are uncertain about the meaning being communicated, and proposed a method (um-RSA) to model this uncertainty within the RSA framework. By using um-RSA to generate predictions about two sources of meaning uncertainty, and testing these predictions with two human experiments, we demonstrated that um-RSA can account for differences in speakers' use of uncertainty expressions that are not predicted by standard RSA models.

Acknowledgments

This work was supported by grants from the Robert H.N. Ho Mind, Brain, and Behavior Initiative at Colgate University. We thank the HSP 2025 audience, Cornell LiMe Lab, Northwestern Experimental Meaning Group and the SEAL Lab, Forrest Davis, Sadhwi Srinivas and Sebastian Schuster for their valuable feedback!

References

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

Judith Degen, Michael Henry Tessler, and Noah D Goodman. 2015. Wonky worlds: Listeners revise world knowledge when utterances are odd. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 37.

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

Camilla Gilmore, Nina Attridge, and Matthew Inglis. 2011. Measuring the approximate number system. *Quarterly Journal of Experimental Psychology*, 64(11):2099–2109.

Noah D. Goodman and Andreas Stuhlmüller. 2013. Knowledge and implicature: Modeling language understanding as social cognition. *Topics in Cognitive Science*, 5(1):173–184.

Justin Halberda, Michèle MM Mazzocco, and Lisa Feigenson. 2008. Individual differences in non-verbal number acuity correlate with maths achievement. *Nature*, 455(7213):665–668.

Michele Herbstritt and Michael Franke. 2019. Complex probability expressions and higher-order uncertainty: Compositional semantics, probabilistic pragmatics and experimental data. *Cognition*, 186:50–71.

Daniel Kahneman and Amos Tversky. 1979. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291.

Justine Kao, Leon Bergen, and Noah Goodman. 2014a. Formalizing the pragmatics of metaphor understanding. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 36.

Justine T. Kao and Noah D. Goodman. 2015. Let's talk (ironically) about the weather: Modeling verbal irony. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 37.

Justine T. Kao, Jean Y. Wu, Leon Bergen, and Noah D. Goodman. 2014b. Nonliteral understanding of number words. *Proceedings of the National Academy of Sciences*, 111(33):12002–12007.

Mel W Khaw, Luminita Stevens, and Michael Woodford. 2021. Individual differences in the perception of probability. *PLoS computational biology*, 17(4):e1008871.

Daniel Lassiter and Noah D. Goodman. 2013. Context, scale structure, and statistics in the interpretation of positive-form adjectives. In *Proceedings of Semantics and Linguistic Theory (SALT)*, volume 23.

Sebastian Schuster and Judith Degen. 2020. I know what you're probably going to say: Listener adaptation to variable use of uncertainty expressions. *Cognition*, 203:104285.

Joakim Sundh. 2024. Human behavior in the context of low-probability high-impact events. *Humanities and Social Sciences Communications*, 11(1).

Thomas S Wallsten, Samuel Fillenbaum, and James A Cox. 1986. Base rate effects on the interpretations of probability and frequency expressions. *Journal of Memory and Language*, 25(5):571–587.

Ilker Yildirim, Judith Degen, Michael K. Tanenhaus, and T. Florian Jaeger. 2016. Talker-specificity and adaptation in quantifier interpretation. *Journal of Memory and Language*, 87:128–143.

Jeffrey Zehr and Florian Schwarz. 2018. Penncontroller for ibex. `https://github.com/PennController/pcibex`. Accessed: 2025-04-26.

A Statistical analyses

A.1 Model details

Model formula prob_utterance $\sim$ Meaning * Event type + (1 + Meaning * Event type | randomid)

Coding of factors

- Meaning: Factor with three levels (10-40%, 50%, 60-90%), resulting in two predictors.
  - Meaning 1: 50% coded as 0; 10-40% coded as 1.
  - Meaning 2: 50% coded as 0; 60-90% coded as 1.
- Event type: Factor with two levels (Election coded as 0, Gumball coded as 1).

A.2 Results from statistical models

Utterance	Coefficient	Estimate	p-value
Not	Meaning 1 (50% vs. 10-40%)	2.897	1.35e-09
	Meaning 2 (50% vs. 60-90%)	-1.582	0.153
	Event type (Gumball vs. Election)	-0.921	0.232
	Meaning 1 : Event type	3.272	2.23e-05
	Meaning 2 : Event type	2.008	0.259
Might	Meaning 1 (50% vs. 10-40%)	0.282	0.3194
	Meaning 2 (50% vs. 60-90%)	-1.415	6.25e-09
	Event type (Gumball vs. Election)	0.982	0.0055
	Meaning 1 : Event type	-4.113	< 2e-16
	Meaning 2 : Event type	-1.835	3.70e-06
Probably	Meaning 1 (50% vs. 10-40%)	-1.683	2.09e-08
	Meaning 2 (50% vs. 60-90%)	2.140	< 2e-16
	Event type (Gumball vs. Election)	-8.561	5.92e-08
	Meaning 1 : Event type	8.177	2.71e-07
	Meaning 2 : Event type	9.249	1.01e-08

Table 2: Results from Experiment 1. A separate model was fit to each utterance.

Utterance	Coefficient	Estimate	p-value
Not	Meaning 1 (50% vs. 10-40%)	10.641	< 2e-16
	Meaning 2 (50% vs. 60-90%)	3.806	< 2e-16
	Event type (Gumball vs. Election)	6.944	< 2e-16
	Meaning 1 : Event type	-6.198	< 2e-16
	Meaning 2 : Event type	-4.270	< 2e-16
Might	Meaning 1 (50% vs. 10-40%)	0.531	0.0114
	Meaning 2 (50% vs. 60-90%)	-1.882	3.12e-11
	Event type (Gumball vs. Election)	-0.210	0.4265
	Meaning 1 : Event type	-0.783	< 0.0114
	Meaning 2 : Event type	0.09413	0.7629
Probably	Meaning 1 (50% vs. 10-40%)	-0.535	0.136
	Meaning 2 (50% vs. 60-90%)	2.711	< 2e-16
	Event type (Gumball vs. Election)	-1.040	0.0433
	Meaning 1 : Event type	0.751	0.2152
	Meaning 2 : Event type	1.046	0.047

Table 3: Results from Experiment 2. A separate model was fit to each utterance.

B Additional results of Experiment 1

Figure 4 replicates Figure 2 with predictions from the uncommitted speaker model.

C Differences between Schuster and Degen (2020) and our experiments

Here are the different ways in which our experimental setup differed from Experiment 1 in Schuster and Degen (2020).

The scene In the scene Schuster and Degen used, the gumball machine was on a table. A little girl who couldn't see the machine tells a man who can see the machine "I want an orange one". The interaction between the child and the man was important for Schuster and Degen because they wanted to eventually study how listeners adapt to different speakers (e.g., change the man in the scene to a woman). Since we only wanted to model the speaker, we did not need such a complex setup. In our scene, the gumball machine was on a table. A person (represneted with a stick figure) who was behind a wall and could not see the machine asked "Will I get an orange gumball?".

Utterance sets Schuster and Degen were interested in collecting participants preferences for a wide range of uncertainty expressions. Therefore, they ran several sub-experiments where they varied the specific utterances used. In each sub-experiment, there were always two uncertainty expressions (e.g., might-probably, probably-not, might-not, not-could, might-could, etc), and a third "other" option. In our work, we were interested in comparing *might*, *probably*, and *not*. We decided to not have an "other" option because it was hard to interpret *why* a participant might choose this option.

Meaning set Schuster and Degen used the following meaning set: 0%, 10%, 25%, 40%, 50%, 60%, 75%, 90%, 100%. In our work we wanted to avoid the 100% because we did not have an "other" option, and thus none of our three utterances could be true. For symmetry, we also excluded the 0%. We kept the meaning set size the same by having more fine-grained increments. Thus our meaning set was: 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%.

Sliders Schuster and Degen asked participants to assign points to each of the utterances, and the sliders automatically jumped back if participants assigned more than 100 points. Our pilot experiments suggested that this task was too tedious and participants were more likely to pick whole numbers (to make the

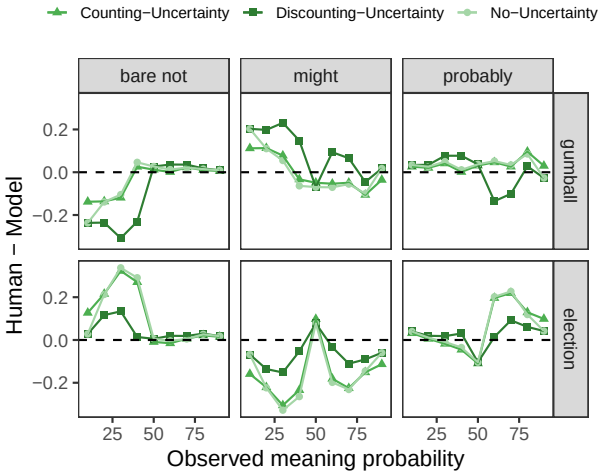


Figure 4: Deviation from uncommitted speaker model predictions averaged over 1000 model instances. Error bars are two SEs.

addition easier). To better capture participants initial instincts, we asked participants to indicate their relative preference and then later normalized these values into probabilities.

C.1 Why was the probability of bare *not* slightly different?

Since Schuster and Degen did not have an experiment where all three of our utterances were present, we selected the *might* and *probably* probabilities from their *might-probably* experiment, and the *bare not* probability from their *might-not* experiment. Since the *might-not* experiment did not have *probably*, the utterance that tends to be most preferred for higher target probabilities, it makes sense that the probability assigned to bare *not* in these cases was higher than in our experiment which did have *probably* as a possible utterance. The close replication in all other cases suggest that the other differences in experimental setup did not impact our results.

D Literal meaning example

	$m_{10\%}$	$m_{20\%}$	$m_{30\%}$	$m_{40\%}$	$m_{50\%}$	$m_{60\%}$	$m_{70\%}$	$m_{80\%}$	$m_{90\%}$
$u_{probably}$	0	0	0	0	0	1	1	1	1
u_{might}	0	0	0	1	1	1	1	1	1
u_{not}	1	0	0	0	0	0	0	0	0

Table 4: Literal meaning map of a hypothetical participant who samples threshold of 10% for not, 30% for might, 60% for probably

E Experimental setup

E.1 Instructions

Instructions in Experiments 1 and 2 were identical except that stimuli were labeled *images* and *scenes*, respectively.

Start of experiment Welcome! In this experiment, you will be presented with images/scenes and be asked to answer questions about each image/scene. For each of A's questions, you will also be presented with three possible responses. Your task is to decide how much you prefer each of the three utterances as a response to A's questions.

Gumball condition In this part of the experiment, you will be presented with images/scenes of gumball machines. The gumball machines are filled with purple and orange gumballs. The gumballs will be tossed around before a random one is dispensed. Here is an example of what a gumball machine may look like.

Election condition In this part of the experiment, you will be presented with images/scenes of election predictions from an unknown country. In this country, the two major parties, Party X and Party Y, compete for votes in different counties. In each county, the party with the maximum votes wins that county. A bipartisan company is interested in studying the probability of the two parties winning in different parts of the country – specifically they are interested in comparing the results in different counties. So they generate predictions about the outcomes of the elections county by county. This company has a great track record of generating very reliable predictions. Here is an example of what a prediction may look like.

Hypothetical listener "A" In this experiment, you will also see a fictional person "A". The fictional person "A", who cannot see the images/scenes, will ask you questions about the images/scenes. Here is an example of what A may look like.

Sliders In the experiment, you will be presented with three possible responses for each of A's questions. For each response, you will use a slider to indicate your degree of preference for the response. Each slider will start at the far left, which indicates zero preference. If you would never use an utterance to answer A's question, leave its slider at the far left. If you would always pick an utterance over the other two utterances, move its slider to the far right and leave the other two sliders in their original positions. If all the sliders are set to the same position, it means you are equally likely to pick any of the utterances to answer A's question.

E.2 Understanding checks

See Figure 5 and 6.

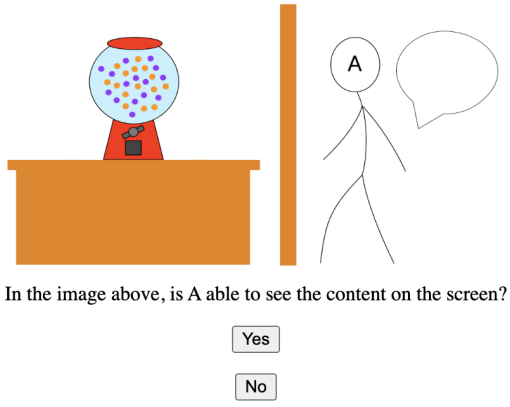


Figure 5: Understanding check for fictional person/ hypothetical listener "A".

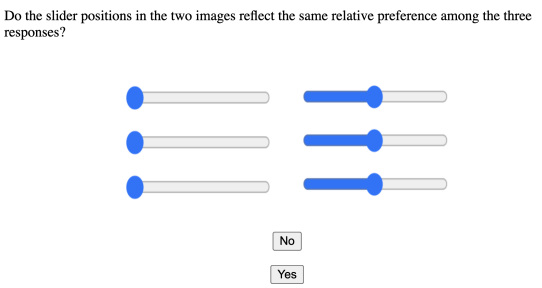


Figure 6: Understanding check for the sliders.

E.3 Example trials

See Figure 7 through 12.

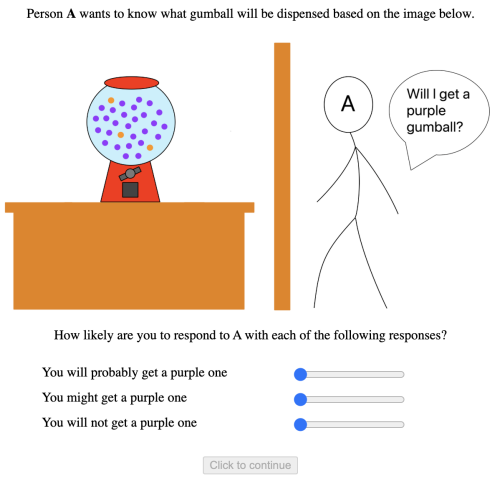


Figure 7: A sample trial of the gumball condition from Experiment 1 (image).

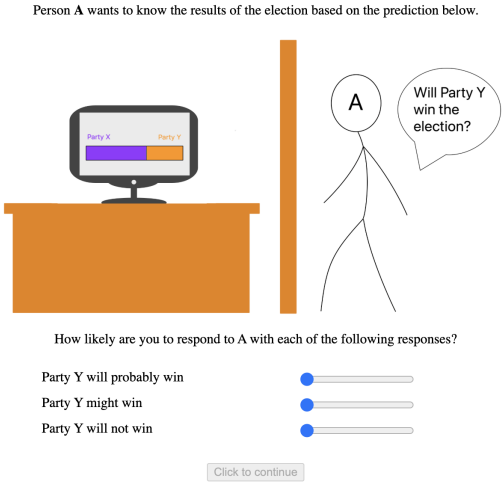


Figure 8: A sample trial of the election condition from Experiment 1 (image).

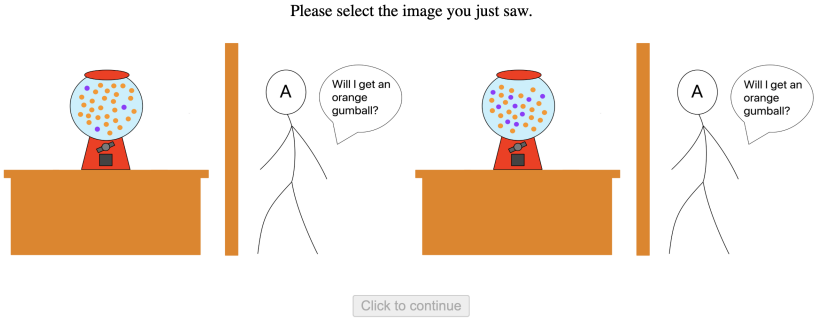


Figure 9: A sample attention check from Experiment 1 (image).

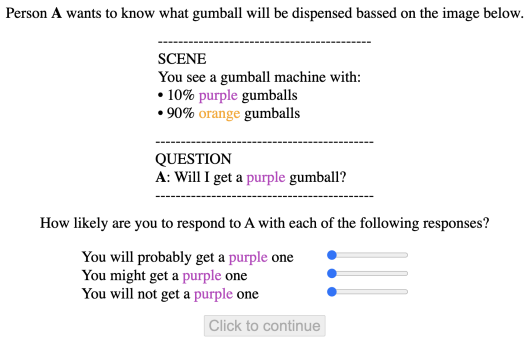


Figure 10: A sample trial of the gumball condition from Experiment 2 (text).

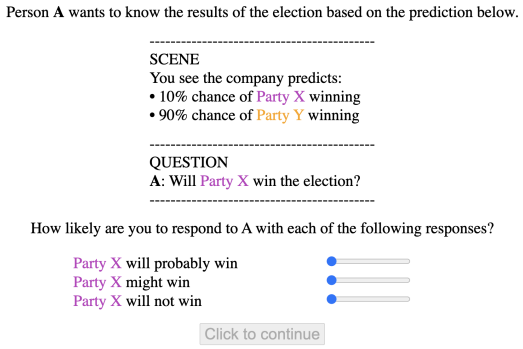


Figure 11: A sample trial of the election condition from Experiment 2 (text).

Figure 12: A sample attention check from Experiment 2 (text).

Learning tone sandhis in Structural Optimality

Izabel Ilie[1] **Andrew Lamont**[1] **Brandon Prickett**[2]
[1]University College London [2]University of Massachusetts Amherst
alexandra.ilie.23@alumni.ucl.ac.uk andrew.lamont@ucl.ac.uk
bprickett@umass.edu

Abstract

This paper examines the learnability of different types of tone sandhi in Structural Optimality (Mortensen, 2006), a constraint-based framework that posits hierarchical scales and defines constraints over the scales. Approached as a hidden structure problem, we show that Expectation Driven Parameter Learning (Jarosz, 2015) can acquire these grammars, but that their properties can make learning difficult.

1 Introduction

Learning an Optimality Theoretic grammar (Prince and Smolensky, 1993/2004) entails determining the ranking of the constraints. When the underlying and surface representations are provided, learning can be achieved using established algorithms (Tesar, 1995; Magri, 2009; Brasoveanu and Prince, 2011). However, it is often the case that surface forms are consistent with multiple, unobservable, structural representations known collectively as hidden structure. Hidden structure complicates the learning problem by requiring the learner to use their grammar to posit an unambiguous surface representation and use it to learn constraint rankings (Tesar, 1998; Tesar and Smolensky, 1998a,b, 2000). Difficulties may compound when different structural parses imply different rankings. For example, the English (eng, Indo-European) word *banana* has medial stress and is consistent with a left-aligned iamb [(.pə.ˈnæ.)ɾə.] or a right-aligned trochee [.pə.(.ˈnæ.ɾə.)]. Adopting Kager's (1999, ch. 4) metrical constraints, the former is consistent with the rankings RhType=I ≫ RhType=T and All-Ft-L ≫ All-Ft-R, as the tableau in (1) illustrates, while the latter is consistent with the opposite rankings RhType=T ≫ RhType=I and All-Ft-R ≫ All-Ft-L.

(1) *banana* with iambic footing

/pənænə/	RhType=I	RhType=T	All-Ft-L	All-Ft-R
☞ [(.pə.ˈnæ.)ɾə.]		1		1
[(.ˈpə.næ.)ɾə.]	W 1	L		1
[.pə.(.næ.ˈɾə.)]		1	W 1	L

This paper examines a different type of hidden structure which is associated globally with the grammar rather than with individual strings. Specifically, we investigate the learnability of Structural Optimality (Mortensen, 2006) which entails learning a language-specific constraint ranking as well as a language-specific scale that organizes elements hierarchically. As in the metrical example, any two mappings may imply contradictory constraint rankings or scales. Our experiments reveal that these grammars are learnable, but that successful acquisition of the correct grammar is not guaranteed. Structural Optimality is illustrated in Section 2, our learning approach is presented in Section 3, and the experiments are reported in Section 4. We discuss the results and plans for future work in Section 6.

2 Structural Optimality

Structural Optimality (Mortensen, 2006) is a theory couched in Optimality Theory (OT; Prince and Smolensky, 1993/2004) that organizes sets of segments into language-specific, hierarchical scales and defines phonotactic and (anti-)faithfulness constraints over the scales. One empirical benefit of Structural Optimality is its ability to model chain shifts including circular chain shifts which are otherwise difficult to model in OT (Moreton, 1999, 2004), but not impossible (Baković, 2007; Łubowicz, 2011; Reiss, 2023). Another is its ability to represent phonetically or phonologically arbitrary scales (Mortensen, 2006, 2.2.1).

To illustrate Structural Optimality, consider tone sandhi in Seenku (sos, Niger-Congo); we follow

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 329–337
July 3-4, 2026 ©2026 Association for Computational Linguistics

McPherson in transcribing nasalization with a tilde below vowels to avoid clashing with toneal diacritics. The language contrasts four phonemic tones: superhigh (S) as in [sí̋] '*Terminalia* sp.', high (H) as in [sí] 'RECIPROCAL', low (L) as in [sì] 'first son (proper name)', and extra low (X) as in [sï̀] 'water jar' (McPherson, 2020, 80). After high-toned pronominal arguments, tone sandhi is circular X → S → H → X, as in the examples in (2). After extra low toned arguments, tone sandhi exhibits a four step chain shift S → H → X → L (3). After super high toned arguments, tones neutralize to super high {X, H, S} → S (4). There are no underlyingly low toned morphemes in the set of sandhi triggers or undergoers (McPherson, 2019, 7).

(2) Circular tone sandhi after /H/ (McPherson, 2019, 8)

X → S	/á kn̰/	[á kn̋]	'your head'
S → H	/á nɨ̋/	[á ní]	'your father'
H → X	/á ɲá/	[á ɲà]	'your mother'

(3) Chain shift tone sandhi after /X/ (McPherson, 2019, 8)

S → H	/à nɨ̋/	[à ní]	'his/her father'
H → X	/à ɲá/	[à ɲà]	'his/her mother'
X → L	/à kn̰/	[à kn̏]	'his/her head'

(4) Neutralizing tone sandhi after /S/ (McPherson, 2019, 8)

X → S	/mɨ́ kn̰/	[mɨ́ kn̋]	'our head'
H → S	/mɨ́ ɲá/	[mɨ́ ɲá]	'our mother'
S → S	/mɨ́ nɨ́/	[mɨ́ nɨ́]	'our father'

In the Structural Optimality analysis,[1] these four tones are organized into the scale {S} < {H} < {X} < {L}, with S at the bottom, H above it, X above H, and L at the top. Each position is represented as a set to allow multiple objects to share a given position in the scale. As mentioned at the beginning of this section, scales can be phonetically or phonologically arbitrary; the scale for Seenku reflects this, as it does not correlate with the relative pitch of the tones S > H > L > X.

The analysis requires five constraints, three Structural Optimality constraints and two general phonotactic constraints. We set aside the question of how to restrict these alternations to the sandhi environments and take for granted some responsible mechanism in the grammar. The three relevant Structural Optimality constraints are defined in (5-7); we define the remaining constraints later in this section. As a simplification, we only consider grammars with single scales; Structural Optimality allows languages to employ multiple scales and defines these constraints relative to specific scales. ENDMOST is a phonotactic constraint that gradiently penalizes tones not at the bottom of the scale; i.e., [H] incurs one violation, [X] two, and [L] three. SAME is a faithfulness constraint that penalizes mappings that change steps on the scale. In this case, because each step on the scale is a singleton set, SAME penalizes all unfaithful mappings. HIGHER is an anti-faithfulness constraint that penalizes tones that do not climb the scale. SAME and HIGHER are categorical, assigning at most one violation per tone.

(5) ENDMOST (Mortensen, 2006, 32)

For every tone T, assign one violation for every step on the scale separating T from the bottom of the scale.

(6) SAME (Mortensen, 2006, 48)

For every pair of corresponding input/output tones T_i, T_o, assign one violation if T_i and T_o are not at the same step in the scale.

(7) HIGHER (Mortensen, 2006, 54)

For every pair of corresponding input/output tones T_i, T_o, assign one violation if T_o is not higher on the scale than T_i.

To these constraints, we add *S{H,L,X} which penalizes any non-super high tone immediately following a super high tone and *HL which penalizes a low tone immediately following a high tone. We do not claim that these constraints are useful elsewhere in the phonology of Seenku or cross-linguistically, but include them for the sake of presenting a complete analysis.

As the tableaux in (8-10) illustrate, the ranking {*HL, *S{H,L,X}} ≫ HIGHER ≫ ENDMOST ≫ SAME models the sandhi processes in Seenku. In the chain shift sandhi (8), the constraint HIGHER prevents underlying tones from surfacing faithfully and from descending the scale. Their ascent is kept stepwise by ENDMOST, which only tolerates minimal distance from the bottom of the scale. Ranked below both constraints, SAME disprefers unfaithful mappings, but cannot prevent their surfacing. Circular tone sandhi (9) is driven by the same pressure for tones to ascend the scale, with *HL preventing low tones from surfacing, making extra low tones

[1] This analysis was previous presented as Ilie et al. (2024).

the effective ceiling. For /HX/, no output can optimally improve on HIGHER, and the tone resets to the bottom of the scale, satisfying ENDMOST. Neutralizing sandhi (10) is driven exclusively by the constraint *S{H,L,X} which penalizes any tone other than a super high tone immediately after a super high tone.

(8) Chain shift tone sandhi after /X/

scale = {S} < {H} < {X} < {L}

		HIGHER	ENDMOST	SAME
/XS/	XS	W 1	L	L
	☞ XH		1	1
	XX		W 2	1
	XL		W 3	1
/XH/	XS	W 1	L	1
	XH	W 1	L 1	L
	☞ XX		2	1
	XL		W 3	1
/XX/	XS	W 1	L	1
	XH	W 1	L 1	1
	XX	W 1	L 2	L
	☞ XL		3	1

(9) Circular tone sandhi after /H/

scale = {S} < {H} < {X} < {L}

		*HL	HIGHER	ENDMOST	SAME
/HX/	☞ HS		1		1
	HH		1	W 1	1
	HX		1	W 2	L
	HL	W 1	L	W 3	1
/HS/	HS		W 1	L	L
	☞ HH			1	1
	HX			W 2	1
	HL	W 1		W 3	1
/HH/	HS		W 1	L	1
	HH		W 1	L 1	L
	☞ HX			2	1
	HL	W 1		W 3	1

In addition to the patterns exemplified by Seenku tone sandhi, Structural Optimality also provides a model of mappings Mortensen (2006, 2.7.3) dubs neutralization with bounce-back. An example from Hmong Sha (hmn, Hmong-Mien) is illustrated

(10) Neutralizing tone sandhi after /S/

scale = {S} < {H} < {X} < {L}

		*S{H,L,X}	HIGHER	ENDMOST	SAME
/SX/	☞ SS		1		1
	SH	W 1	1	W 1	1
	SX	W 1	1	W 2	L
	SL	W 1	L	W 3	1
/SH/	☞ SS		1		1
	SH	W 1	1	W 1	L
	SX	W 1	L	W 2	1
	SL	W 1	L	W 3	1
/SS/	☞ SS		1		
	SH	W 1	L	W 1	W 1
	SX	W 1	L	W 2	W 1
	SL	W 1	L	W 3	W 1

in (11); see Mortensen (2006, 4.2-4.3) for other examples. In these examples, tildes below vowels indicate glottalization; see Johnson and Strecker (2020, 15-17) for a detailed description of the tones, their variable realizations, and their associated laryngeal features. Three of the mappings neutralize tones 2, 4, and 8 to tone 6, reminiscent of neutralizing sandhi in Seenku (4). Rather than surfacing faithfully however, tone 6 is mapped onto tone 8, metaphorically bouncing back.

(11) Neutralizing tone sandhi with bounce-back in Hmong Sha (Johnson and Strecker, 2020, 19)

2 → 6	/sṵ˩ mpje˨˩/ ‘ear’	[sṵ˩ mpje̤˦]
4 → 6	/nti˥ tʰɘʉ˨˩ te̤˦/ ‘thumb’	[nti˥ tʰɘʉ˨˩ te̤˦]
8 → 6	/qa˦ naɪ̰˩/ ‘female (hen)’	[qa˦ naɪ̤˦]
6 → 8	/qa˦ nt͡ʃar˦ ma̤˦/ ‘eyeball’	[qa˦ nt͡ʃar˦ ma̰˩]

Neutralization with bounce-back is driven by the anti-faithfulness constraint DIFF (12) which penalizes mappings that are faithful with respect to the scale.

(12) DIFF (Mortensen, 2006, 49)

For every pair of corresponding input/output tones T_i, T_o, assign one violation if T_i and T_o are at the same step in the scale.

The tableaux in (13) illustrate a Structural Optimality analysis of Hmong Sha sandhi with the scale $\{6\} < \{8\} < \{2, 4\}$. In the sandhi environment (see Johnson and Strecker, 2020, 18), after a preceding tone, represented as T, tones 2, 4, and 8 surface as tone 6. The anti-faithfulness constraint DIFF rules out intra-step mappings, and ENDMOST prefers candidates with tone 6, which occupies the bottom of the scale. Because of DIFF, underlying tone 6 cannot surface faithfully. It maps onto tone 8 to minimze the violations of ENDMOST and remain as close as possible to the bottom of the scale.

(13) Neutralizing tone sandhi with bounce-back

scale = $\{6\} < \{8\} < \{2, 4\}$

		DIFF	ENDMOST	HIGHER	SAME
/T2/ /T4/	☞ T6			1	1
	T8		W 1	1	1
	T2	W 1	W 2	1	L
	T4	W 1	W 2	1	L
/T8/	☞ T6			1	1
	T8	W 1	W 1	1	L
	T2		W 2	L	1
	T4		W 2	L	1
/T6/	T6	W 1	L	W 1	L
	☞ T8		1		1
	T2		W 2		1
	T4		W 2		1

For the sake of completeness, we define three additional Structural Optimality constraints in (14-16). With the four constraints defined previously, these seven constraints cover input-output mappings; see Mortensen (2006, 2.5) for related constraints defined over strings.

(14) NOHIGHER (Mortensen, 2006, 55)

For every pair of corresponding input/output tones T_i, T_o, assign one violation if T_o is higher on the scale than T_i.

(15) TOP (Mortensen, 2006, 34)

For every tone T, assign one violation if T is not at the top of the scale.

(16) EXTREME (Mortensen, 2006, 34-35)

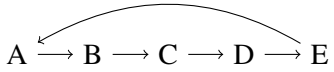
For every tone T, assign one violation if there is a step below T and a step above T.

To summarize, Structural Optimality predicts a modest typology including the five language types abstractly schematized in (17-21) following Mortensen (2006, 2.7). Additional languages where the top and bottom of the scale map onto themselves result from ranking ENDMOST relatively high; we omit these languages from discussion for brevity's sake.

(17) Chain shift

A ⟶ B ⟶ C ⟶ D ⟶ E ⮌

(18) Circular chain shift



(19) Identity

A ⮌ B ⮌ C ⮌ D ⮌ E ⮌

(20) Neutralization

(21) Neutralization with bounce-back

3 Expectation Driven Parameter Learning

Because the scales posited by Structural Optimality cannot be observed directly, they instantiate a hidden structure problem in learning. For example, if a given mapping /α/ → [β] is interpreted as the end of a chain shift, then it provides evidence for ranking HIGHER ≫ ENDMOST and evidence that α is lower on the scale than β as in (8). If the mapping is instead interpreted as neutralization, then it provides evidence for ranking ENDMOST ≫ HIGHER and evidence that β is lower on the scale than α as in (13).

Our experiments employ Expectation Driven Parameter Learning (EDPL; Jarosz, 2015), which has been shown successfully to learn grammars with hidden structure in phonology (Nazarov, 2016; Nazarov and Jarosz, 2017; Prickett and Jarosz, 2021; Nazarov and Jarosz, 2021) and syntax (Prickett et al., 2020; Hucklebridge, 2023), consistently outperforming related models. In a nutshell, EDPL

estimates the probability of some part of the grammar behaving in a particular way by randomly sampling all possible grammars that share that grammatical characteristic; the more accurate the sampled grammars, the more likely it is that that particular aspect has that behavior.

In EDPL, grammars are represented as sets of binary parameters. For example, the relative ranking of every pair of OT constraints is associated with two parameters: ψ(HIGHER, ENDMOST) represents the probability that HIGHER dominates ENDMOST, and ψ(ENDMOST, HIGHER) represents the probability that the opposite ranking holds. It is worth noting that contradictory parameters $\psi(\alpha, \beta)$ and $\psi(\beta, \alpha)$ are not defined as summing to 1, but this is an automatic consequence of learning.

Exactly like OT constraint rankings, scales in Structural Optimality are defined as directed acyclic graphs (Mortensen, 2006, 2.2). Accordingly, parallel binary parameters represent the grammar's scale. For example, ψ(2, 6) represents the probability that tone 2 occupies a lower step on the scale than tone 6, and ψ(6, 2) represents the reverse. Because learning a Structural Optimality grammar entails learning two directed acyclic graphs in parallel, it is the same computational problem as learning a stratal OT grammar (Kiparsky, 2000) with two levels. We thus based our implementation on Prickett and Jarosz (2021), who use EDPL in that context.

As a running example, consider a language with three tones, L, M, H, that participate in a chain shift L $\rightarrow$ M $\rightarrow$ H$\circlearrowleft$. As a simplification, there are only two constraints HIGHER and ENDMOST. For each pair of tones, there are two parameters representing their relative position on the scale, and there are two parameters representing the rankings of the two constraints. At the beginning of learning, these parameters can be randomized or set uniformly to 0.5 as illustrated below.

(22) Initial state of learning

ψ(L, M) = 0.5 $\quad$ ψ(M, L) = 0.5
ψ(L, H) = 0.5 $\quad$ ψ(H, L) = 0.5
ψ(M, H) = 0.5 $\quad$ ψ(H, M) = 0.5
ψ(ENDMOST, HIGHER) = 0.5
ψ(HIGHER, ENDMOST) = 0.5

Following Jarosz (2015, 15), the probability that a given parameter has a given value ψ_i given the current state of the grammar G and a training datum d is defined using Bayes' Law in (23). Computing the probability of the grammar generating the datum with this parameter setting $P(d|\psi_i, G)$ and the current probability of the parameter setting given the grammar $P(\psi_i|G)$ are straightforward. To compute the probability of the datum given the grammar $P(d|G)$, however, requires marginalizing over all possible parameter settings, which is intractable: there are 2^n possible grammars given binary parameters over n elements. Instead, this value is estimated by randomly sampling possible grammars according to the model's current estimate of those parameter probabilities. Sampled grammars' constraint rankings and scales were converted into a total order using topological sorting (Kahn, 1962).

(23) $P(\psi_i|G, d) = \frac{P(d|\psi_i,G)P(\psi_i|G)}{P(d|G)}$

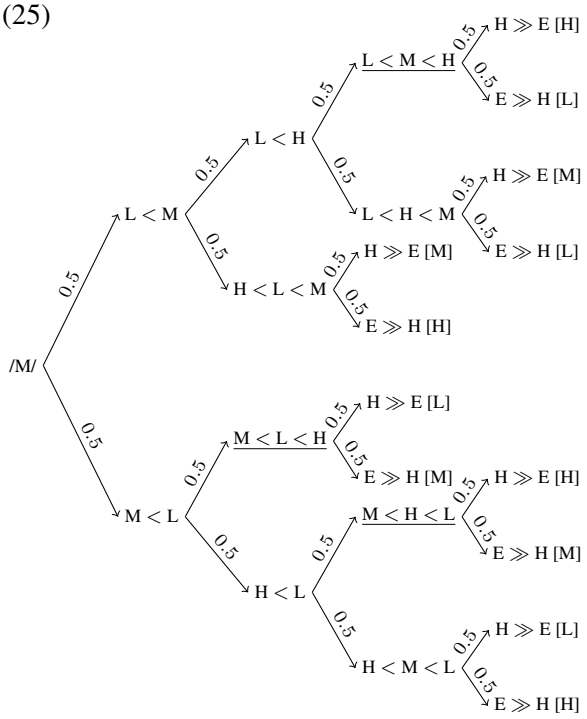
In the running example, consider the parameter ψ(M, H), that M is lower on the scale than H, and the datum /M/ $\rightarrow$ [H]. Because the number of parameters is relatively small, the entire grammar space can be represented in (25) on the next page. In the upper branch, L is lower on the scale than M, and these grammars occupy half of the probability space because ψ(L, M) = 0.5. These grammars further branch into those in which L is also lower on the scale than H or the opposite holds. In the latter, the scale parameters are all set: L is below M, H is below L, and by transitivity, H must be below L as well. At the end of each branch, the relative ranking of ENDMOST and HIGHER is represented. The surface form of /M/ is labelled at the end of each branch. Overall, the grammar correctly maps /M/ onto [H] with probability 0.3125, incorrectly produces [L] with probability 0.3125, and [M] with probability 0.375. The slight boost in faithful mappings correspond to M occupying the top or bottom of the scale. Marginalizing over grammars where M is lower on the scale than H (underlined), the probability of /M/ $\rightarrow$ [H] is 0.125. Therefore the probability of the parameter setting ψ(M, H) given this state of the grammar and the mapping /M/ $\rightarrow$ [H] is $\frac{0.125 \times 0.5}{0.3125} = 0.2$.

As the grammar space explodes in the number of parameters, the probabilities of data given the grammar (and a parameter setting) cannot be efficiently calculated and are instead estimated by randomly sampling from the possible grammars. For Structural Optimality, the space of grammars grows both in the number of constraints and the size of the scales; as far as we know, this is a unique aspect of the framework. When sampling, to calculate the probability of some datum given the grammar, we

have to marginalize over the opposite parameter setting. Thus the denominator in (23) expands to (24) following Jarosz (2015, 15). By $\neg\psi_i$, we mean the opposite parameter setting to ψ_i, for example $\neg\psi$(M, H) is ψ(H, M).

(24) $P(d|G) = P(d|\psi_i, G)P(\psi_i|G) + P(d|\neg\psi_i, G)P(\neg\psi_i|G)$

(25)



In our experiments, we estimate parameter settings using online learning, meaning that incremental updates are calculated once per data point. Online learning is parameterized by a sampling rate r and a learning rate p; in our experiments, we set r to 100 and p to 0.75. Following Jarosz (2015, 30), we operationalize online learning as follows:

(26) For a UR→SR mapping and a parameter setting ψ(A, B),

- Set A ≫ B (if A and B are constraints) or A < B (if A and B are elements on the scale) and sample 100 grammars. m_{AB} is the number of times the UR is correctly mapped onto its SR.
- Set B ≫ A or B < A and sample 100 grammars. m_{BA} is the number of times the UR is correctly mapped onto its SR.
- The expected number of correct mappings $\hat{E}_{AB}$ and $\hat{E}_{BA}$ are the number of correct mappings times the probability of that parameter setting
  $\hat{E}_{AB} = m_{AB} \times \psi$(A, B)
  $\hat{E}_{BA} = m_{BA} \times \psi$(B, A)
- The new value of the parameter setting ψ(A, B) is the weighted sum of its current value and expected number of correct mappings normalized by the two settings.
  ψ_{new}(A, B) = $0.75 \times \frac{\hat{E}_{AB}}{\hat{E}_{AB}+\hat{E}_{BA}} + 0.25 \times \psi_{current}$(A, B)
  Likewise for ψ(B, A).
  ψ_{new}(B, A) = $0.75 \times \frac{\hat{E}_{BA}}{\hat{E}_{AB}+\hat{E}_{BA}} + 0.25 \times \psi_{current}$(B, A)

Consider again the mapping /M/ → [H] and the parameter ψ(M, H) in the initial grammar. With 100 samples, we expect to sample 25 correct mappings with M < H and about 38 with H < M. With the learning rate $p = 0.75$, ψ(M, H) is lowered from 0.5 to 0.423, approaching its calculated value of 0.2, and ψ(H, M) is raised from 0.5 to 0.577. Because the actual grammar needed to model the chain shift has the scale L < M < H and the ranking HIGHER ≫ ENDMOST, this mapping incorrectly leads the learner towards a neutralization grammar with H at the bottom of the scale and the opposite constraint ranking. This makes sense because neutralization grammars are consistent with more scales than chain shift grammars and therefore occupy more of the grammar space. The mapping /L/ → [M] is inconsistent with a neutralization grammar and will push the parameter settings away from it over the course of learning. Success of the learner therefore depends on these data points working together to pull the grammar in the correct direction when their effects are averaged over the course of acquisition.

4 Experiments

Our experiments test EDPL on the five language types in (17-21) with two, three, four, and five segments each. Learning was online over 20 epochs of 100 sampled data points each. Initial pilots did not indicate much improvement with additional epochs or a higher sampling rate. All parameters were initialized to 0.5 and the learning rate was set to 0.75.[2]

[2]Our software and training data are available at https://github.com/aphonologist/EDPL_StructuralOT.

Experiments consisted of a set of input/output mappings that abstracted away from phonological substance: segments are represented by the letters A, B, etc. For example, a three segment language with a circular chain shift is represented as the set of input/output pairs {(A,B), (B,C), (C,A)}. For each input, every segment was a possible candidate, and the constraint set consisted of the Structural Optimality constraints defined in Section 2. Each run of the learner was assigned an accuracy score by averaging across the probabilities it assigned to all correct mappings. In the circular chain shift example, if /A/ mapped onto [B] with probability 0.5, /B/ onto [C] with probability 0.25, and /C/ onto [A] with probability 0.0, the learner's accuracy would be 0.25. Because EDPL is non-deterministic, we report average accuracies over 20 separate runs in each condition.

The experimental results are summarized in (27). Overall, the learner was successful, although it struggled with chain shifts defined over four segments and neutralization with bounce-back over three segments. Languages defined over two segments are ambiguous and we arbitrarily group them into categories. For example, {(A,B), (B,B)} is both a chain shift and neutralization.

(27) Summary of results by language and number of elements

	Chain shift	Circular chain shift	Identity	Neutralization	Neutralization with bounce-back
2	94.5%	87.0%	76.5%	N/A	N/A
3	63.0%	76.9%	80.1%	89.3%	47.7%
4	47.5%	77.0%	87.0%	89.8%	70.9%
5	62.9%	78.3%	92.3%	91.5%	66.8%

5 Understanding the Model's Performance

There is a trend in (27) where circular chain shifts, identity mappings, and neutralization mappings are more accurate with larger numbers of segments. For all three languages, there are multiple scales that are consistent with them. For example, the circular chain shift A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ A can be defined with three scales A $<$ B $<$ C, C $<$ A $<$ B, and B $<$ C $<$ A, and in general, a circular chain shift over n segments is consistent with n scales. Neutralization mappings over n segments are consistent with $(n-1)!$ scales: the output is fixed at the bottom of the scale and the remaining $n-1$ segments can be in any order. Similarly, identity mappings do not use the scales at all, and are consistent with all $n!$ scales. By contrast, chain shifts can only be modeled with one scale.

It appears that EDPL is more successful modeling languages that are consistent with more grammars – i.e., more constraint rankings and scales for more evidence of this bias in EDPL, see Jarosz, 2016, Nazarov and Jarosz, 2021, and Prickett, 2021, 57-59. To measure this, we ran a factorial typology and calculated the R-volume of each language. R-volume is simply the total number of rankings that model a given language (Bane and Riggle, 2008; Riggle, 2010); for Structural Optimality, we also include the number of scales. The table in (28) reports the R-volumes for each of the languages in our experiments relative to the total number of grammars for that number of segments. There is a strong positive correlation (Pearson's $r = 0.677$) between learning success and R-volume suggesting that our interpretation is on the right track.

(28) Relative R-volume

	Chain shift	Circular chain shift	Identity	Neutralization	Neutralization with bounce-back
2	25.0%	25.0%	25.0%	N/A	N/A
3	0.2%	2.4%	17.9%	16.3%	3.4%
4	0.0%	0.8%	17.9%	12.3%	1.7%
5	0.0%	0.2%	17.9%	9.8%	1.0%

When the learner failed, it appeared to get stuck primarily in languages with larger R-volumes. For chain shift inputs, 24% of simulations resulted in neutralization languages and 7% resulted in circular chain shifts. 18% of circular chain shift inputs resulted in neutralization mappings. The only exceptions to this trend are the 20% of identity language inputs that resulted in neutralization mappings and the 2% that resulted in circular chain shifts.

Another possible contributing factor to the difficulty of each language is the availability and ambiguity inherent in mappings (for more on this in the context of EDPL, see Prickett et al., 2020 and Hucklebridge, 2023). Given a set of elements, every mapping /α/ $\rightarrow$ [β] is attested in the predicted

typology. For example, with β at the bottom of the scale, there is a ranking that neutralizes all segments to it. Further, the number of languages grows with the number of elements, which is unique to Structural Optimality. There are 12 predicted languages with two elements, 72 with three elements, 177 with four elements, and 570 with five elements.

Within languages, mappings are maximally ambiguous. The ambiguity of the footing of the word *banana* in Section 1 disappears once a learner encounters longer words. Intuitively, fewer grammars produce penultimate stress in *jalapeño* than in *banana*. For the Structural Optimality typology, mappings do not exhibit any such asymmetries; they are evenly dispersed in different languages both in their relative frequencies and their ranking volumes (Bane and Riggle, 2008; Riggle, 2010). Only faithful mappings are slightly under-represented: with four elements, each faithful mapping has an R-volume of 14 and an overall frequency of 0.237 compared to unfaithful mappings which have R-volumes of 15 and frequencies of 0.254 and with five elements, each faithful mapping has an R-volume of 34 (vs. 39) and a frequency of 0.179 (vs. 0.205).

Another factor contributing to difficulty of learning is the capacity for scales to be arbitrary: there is no inherent preference for any given scale. We hypothesize that including a prior for scales to be phonetically or phonologically coherent (for example, by initializing the probability of certain scale parameter settings to values other than 0.5) may improve learning by pushing the model away from certain, phonetically unmotivated grammars for an example of this kind of bias using a different approach to learning, see White, 2013 and Hayes and White, 2015. We leave for future work the question of whether imposing such a naturalness bias aids learning.

6 Discussion

In general, we found that Structural Optimality grammars were learned more often than not by our model, but that acquisition of these languages was not always guaranteed. Other learning algorithms, such as L-BFGS-B (Byrd et al., 1995) have also been adapted for hidden structure in phonology (Lee et al., 2026) and stratal, constraint-based grammars (Nazarov and Pater, 2017), so future work should explore other approaches to see if they result in higher accuracies.

Future work should also compare the results of our learning model to human behavior and typological trends. If languages that the model acquires with high accuracy correspond to languages that are easier for humans to learn or that are more common typologically, this would be strong evidence for this approach to learning hidden structure.

For now, these results show another useful application of EDPL (Jarosz, 2015) and represent a promising first step in modeling the acquisition of tonal scales.

Acknowledgments

We are grateful to three anonymous reviewers and Gaja Jarosz for their valuable feedback. All remaining errors are of course our own.

References

Eric Baković. 2007. A revised typology of opaque generalisations. *Phonology*, 24(2):217–259.

Max Bane and Jason Riggle. 2008. Three correlates of the typological frequency of quantity-insensitive stress systems. In *Proceedings of the Tenth Meeting of ACL Special Interest Group on Computational Morphology and Phonology*, pages 29–38. The Association for Computational Linguistics.

Adrian Brasoveanu and Alan Prince. 2011. Ranking and necessity: the Fusional Reduction Algorithm. *Natural Language & Linguistic Theory*, 29(1):3–70.

Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208.

Bruce Hayes and James White. 2015. Saltation and the P-map. *Phonology*, 32(2):267–302.

Shay Hucklebridge. 2023. Learning and the typology of word order: a model of the Final-over-Final Condition. *Glossa: a journal of general linguistics*, 8(1).

Izabel Ilie, Juliette van Steensel, and Andrew Lamont. 2024. Sticking point: Structural Optimality cannot model tone bunnies. Poster presented at *LAGB 2024*.

Gaja Jarosz. 2015. Expectation driven learning of phonology. *Unpublished manuscript, University of Massachusetts Amherst. Available at* `https://people.umass.edu/jarosz/edl_submitted.pdf`.

Gaja Jarosz. 2016. Learning opaque and transparent interactions in Harmonic Serialism. In *Proceedings of the 2015 Annual Meeting on Phonology*, Washington, D.C. Linguistic Society of America.

Michael Johnson and David Strecker. 2020. An overview of Hmong Sha (West Hmongic, Guangnan) Phonology and Lexicon. Unpublished manuscript. Available at https://www.academia.edu/44730663/An_Overview_of_Hmong_Sha_West_Hmongic_Guangnan_Phonology_and_Lexicon.

René Kager. 1999. *Optimality Theory*. Cambridge Textbooks in Linguistics. Cambridge University Press, Cambridge.

A. B. Kahn. 1962. Topological sorting of large networks. *Communications of the ACM*, 5(11):558–562.

Paul Kiparsky. 2000. Opacity and cyclicity. *The Linguistic Review*, 17:351–365.

Seung Suk Lee, Joe Pater, and Brandon Prickett. 2026. Representing and learning stress: A MaxEnt framework for comparing learning across grammatical theories. Unpublished manuscript, University of Massachusetts Amherst. Available at https://websites.umass.edu/pater/papers/.

Anna Łubowicz. 2011. Chain shifts. In Marc van Oostendorp, Colin Ewen, Elizabeth Hume, and Keren Rice, editors, *The Blackwell Companion to Phonology*, pages 1717–1735. Wiley-Blackwell, Malden, MA.

Giorgio Magri. 2009. *A Theory of Individual-Level Predicates Based on Blind Mandatory Implicatures. Constraint Promotion for Optimality Theory*. Ph.D. thesis, Massachusetts Institute of Technology.

Laura McPherson. 2019. Seenku argument-head tone sandhi: Allomorph selection in a cyclic grammar. *Glossa*, 4(1).

Laura McPherson. 2020. *A Grammar of Seenku*, volume 83 of *Mouton Grammar Library*. De Gruyter Mouton, Berlin/Boston.

Elliott Moreton. 1999. Non-computable functions in Optimality Theory. University of Massachusetts Amherst manuscript.

Elliott Moreton. 2004. Non-computable functions in Optimality Theory. In John J. McCarthy, editor, *Optimality Theory in Phonology: A Reader*, pages 141–163. Blackwell Publishing, Malden, MA.

David Mortensen. 2006. *Logical and Substantive Scales in Phonology*. Ph.D. thesis, University of California, Berkeley.

Aleksei Nazarov. 2016. *Extending Hidden Structure Learning: Features, Opacity, and Exceptions*. Ph.D. thesis, University of Massachusetts Amherst.

Aleksei Nazarov and Gaja Jarosz. 2017. Learning parametric stress without domain-specific mechanisms. In *Proceedings of the 2016 Annual Meeting on Phonology*, Washington, D.C. Linguistic Society of America.

Aleksei Nazarov and Gaja Jarosz. 2021. The Credit Problem in parametric stress: A probabilistic approach. *Glossa: a journal of general linguistics*, 6(1).

Aleksei Nazarov and Joe Pater. 2017. Learning opacity in Stratal Maximum Entropy Grammar. *Phonology*, 34(2):299–324.

Brandon Prickett. 2021. *Learning Phonology with a Sequence-to-Sequence Neural Network*. Ph.D. thesis, University of Massachusetts Amherst.

Brandon Prickett, Kaden Holladay, Shay Hucklebridge, Max Nelson, Rajesh Bhatt, Gaja Jarosz, Kyle Johnson, Aleksei Nazarov, and Joe Pater. 2020. Learning syntactic parameter settings without triggers by assigning credit and blame. In *Proceedings of the Fifty-fifth Annual Meeting of the Chicago Linguistic Society*, pages 337–350, Chicago, IL. Chicago Linguistic Society.

Brandon Prickett and Gaja Jarosz. 2021. Modeling the acquisition of phonological interactions: Biases and generalization. In *Supplemental Proceedings of the 2020 Annual Meeting on Phonology*, Washington, D.C. Linguistic Society of America.

Alan Prince and Paul Smolensky. 1993/2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell Publishing, Malden, MA.

Charles Reiss. 2023. A simple account of chain shifts and saltations in Optimality Theory. Unpublished manuscript, Concordia University.

Jason Riggle. 2010. Sampling rankings. Unpublished manuscript, University of Chicago.

Bruce Tesar. 1995. *Computational Optimality Theory*. Ph.D. thesis, University of Colorado.

Bruce Tesar. 1998. An iterative strategy for language learning. *Lingua*, 104(1-2):131–145.

Bruce Tesar and Paul Smolensky. 1998a. Learnability in Optimality Theory. *Linguistic Inquiry*, 29(2):229–268.

Bruce Tesar and Paul Smolensky. 1998b. Learning Optimality-Theoretic grammars. *Lingua*, 106(1-4):161–196.

Bruce Tesar and Paul Smolensky. 2000. *Learnability in Optimality Theory*. The MIT Press, Cambridge, MA.

James White. 2013. *Bias in Phonological Learning: Evidence from Saltation*. Ph.D. thesis, University of California, Los Angeles.

Making "concreteness" more concrete

Yanting Li, Gregory Scontras, and Richard Futrell
Department of Language Science
University of California, Irvine
{yantil5, g.scontras, rfutrell}@uci.edu

1 Introduction

Concrete words (e.g., *apple* as opposed to *appetite*) are often described as more specific (Paivio, 1963) and perceptible (Lambert, 1955), with a more direct reference to objects, material, or sources of sensation (Gorman, 1961; Paivio, 1966). Concreteness is an important concept in the field of psycholinguistics. Previous research has found that concrete words are generally more easily learned (Paivio, 1963), more easily retrieved for lexical decision tasks (Schwanenflugel and Akin, 1994), and they tend to elicit more associated words (Lambert, 1955) with shorter response time (Richardson, 1975; Paivio, 1966). Different theories have been proposed to explain this concreteness effect. The Dual Coding Theory proposes that concrete words have nonverbal (perceptual) representations in addition to verbal ones, facilitating processing (Paivio, 1986). The Context Availability Theory, on the other hand, proposes that concrete words have more relevant contexts in people's prior knowledge that aid processing (Schwanenflugel et al., 1988).

In bilingual populations, concrete words have been shown to elicit faster lexical decision response times in both languages (van Hell and de Groot, 1998a) and elicit more similar lexical associations across languages than abstract words (Taylor, 1976; van Hell and de Groot, 1998b). Additionally, they have been found to facilitate code-switching (Myslín and Levy, 2015). One explanation is that concrete words share more semantic features across languages, resulting in more similar cross-linguistic associations and easier co-activation of translation equivalents (van Hell and de Groot, 1998b; Kroll et al., 2006). Abstract words, by contrast, are more dependent on the linguistic context in which they appear rather than on perceptual properties (Breedin et al., 1994), making them more likely to have language-specific meanings and trigger different associations across languages (van Hell and de Groot, 1998b).

Are concrete words more similar across languages than abstract words? Multilingual aligned word vectors offer a useful way to test this question, as they locate words from different languages in a shared embedding space by learning the contexts in which they typically occur (Mikolov et al., 2013; Smith et al., 2017; Conneau et al., 2017). In this shared space, distance between any two vectors can indicate their semantic relatedness. If concrete words do share more semantic features across languages, they should be located closer to their nearest neighbor in another language than abstract words. This is the hypothesis we aim to test.

2 Method

Concreteness ratings We found previously-collected concreteness ratings of words in six languages: Dutch (Brysbaert et al., 2014a), English (Brysbaert et al., 2014b), French (Bonin et al., 2018), Cypriot Greek (Traikapi et al., 2024), Mandarin (Xu and Li, 2020) and Portuguese (Soares et al., 2017) (Table 1). These languages were chosen because they are among the 44 languages that have pre-trained aligned word vectors (Joulin et al., 2018; Bojanowski et al., 2017). For all six datasets, concreteness was rated by self-identified adult native speakers recruited from universities or through crowd-sourcing platforms. "Concrete word" were generally described as words that can be experienced, perceived or learned through the senses. Each word was typically shown in isolation without linguistic context[1] and received around 30 rat-

[1]This could introduce the problem of "concreteness ambiguity" for lexical items with more than one meaning that have different degrees of concreteness (Löhr, 2021). For example, *table* referring to a piece of furniture is relatively concrete, whereas *table* referring to a set of values in a document may be less so. However, pretrained word embeddings also bear such ambiguity, as the embedding of a word form reflects the aggregate of all contexts it appears in. We therefore do not expect such ambiguity to systematically bias the relationship

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 338–342
July 3-4, 2026 ©2026 Association for Computational Linguistics

Language	Total word count	Words with vector	Word information
Dutch	30,070	28,095	lemmas from all syntactic categories
English	39,954	35,318	lemmas from all syntactic categories
French	1,659	1,576	67% nouns, plus verbs, adjectives, adverbs and function words
Cypriot Greek	2,948	327	mostly nouns, with a few adjectives and verbs
Mandarin	9,877	8,421	two-character simplified Chinese words, excluding prepositions
Portuguese	3,800	3,770	content words including nouns, adjectives and others

Table 1: Concreteness rating information for the six languages used in the study.

ings, with counts ranging from 15 to 62 across languages.

Vector distance For each word with a rating, we located it in the 44-language embedding space with pre-trained multilingual aligned word vectors (Joulin et al., 2018; Bojanowski et al., 2017) and measured both the Euclidean distance and cosine similarity to its nearest neighbor among words in the other 43 languages.[2] We focused on nearest neighbors rather than translation equivalents because the latter is often unavailable or unstable for abstract words (van Hell and de Groot, 1998b). Even given a concrete word, there might be multiple translation equivalents. Therefore, identifying and determining translation equivalents for each word across 44 languages would require substantial language-specific judgment and introduce additional subjectivity into the analysis. Using the nearest neighbor instead could provide a consistent and fully data-driven approximation of cross-linguistic semantic similarity. As Euclidean distance and cosine similarity between two vectors were strongly correlated, analyses based on either metric yielded similar results. In the remainder of the paper, we report results based on Euclidean distance.

Word frequency Word frequency may influence both concreteness ratings and vector distances. For example, function words tend to be both highly frequent and relatively abstract; more frequent words generally have more stable embedding representations. We therefore included word frequency in our analyses. The English concreteness rating dataset included frequency information from SUBTLEX-US (Brysbaert and New, 2009). To maintain consistency across languages, we added frequency information for Dutch and Mandarin using SUBTLEX-NL (Keuleers et al., 2010) and SUBTLEX-CH (Cai and Brysbaert, 2010), respectively. For the remaining three languages for which SUBTLEX-based frequency norms were not available, log frequency estimates were obtained using the `wordfreq` Python library (Speer, 2022).

Analysis For each language with concreteness ratings, we fit the following linear regression model 43 times, each time using a different language as the source of nearest-neighbor distances:

`concreteness` $\sim$ `distance + word frequency`

3 Results

Fig 1 illustrates the relationship between word concreteness ratings and the distance to the nearest neighbor across 44 languages. The six panels represent the six languages for which concreteness ratings are available, and each line within a panel corresponds to one of the 43 neighboring languages. Solid lines indicate that distance was a significant predictor in the regression model described above ($p < .05$), whereas dashed lines indicate non-significant effects. The relatively lower proportion of significant effects observed for Cypriot Greek (91%) and French (44%) may partly reflect smaller effective sample sizes, as these datasets are relatively limited to begin with and the analyses additionally require the availability of pretrained aligned embeddings for each word.

The relationship between concreteness rating and distance varies considerably across languages. For English and Mandarin, the correlations are uniformly negative and significant across all neighboring languages. For French, the correlation is significant for 19 of the 43 neighboring languages, and all significant effects are negative. In contrast, Dutch, Cypriot Greek, and Portuguese generally exhibit positive correlations, with a few exceptions (Chinese and Vietnamese for Portuguese; and additionally Arabic for Dutch).

between concreteness and cross-linguistic semantic distance.

[2]Because aligned word vectors are trained on large multilingual corpora, the vocabulary of a given language may contain words from other languages. To minimize such cross-language contamination, we restricted searches to words written in the dominant script of the target language. For example, when Bulgarian was the neighboring language, the nearest neighbor must be written in Cyrillic script.

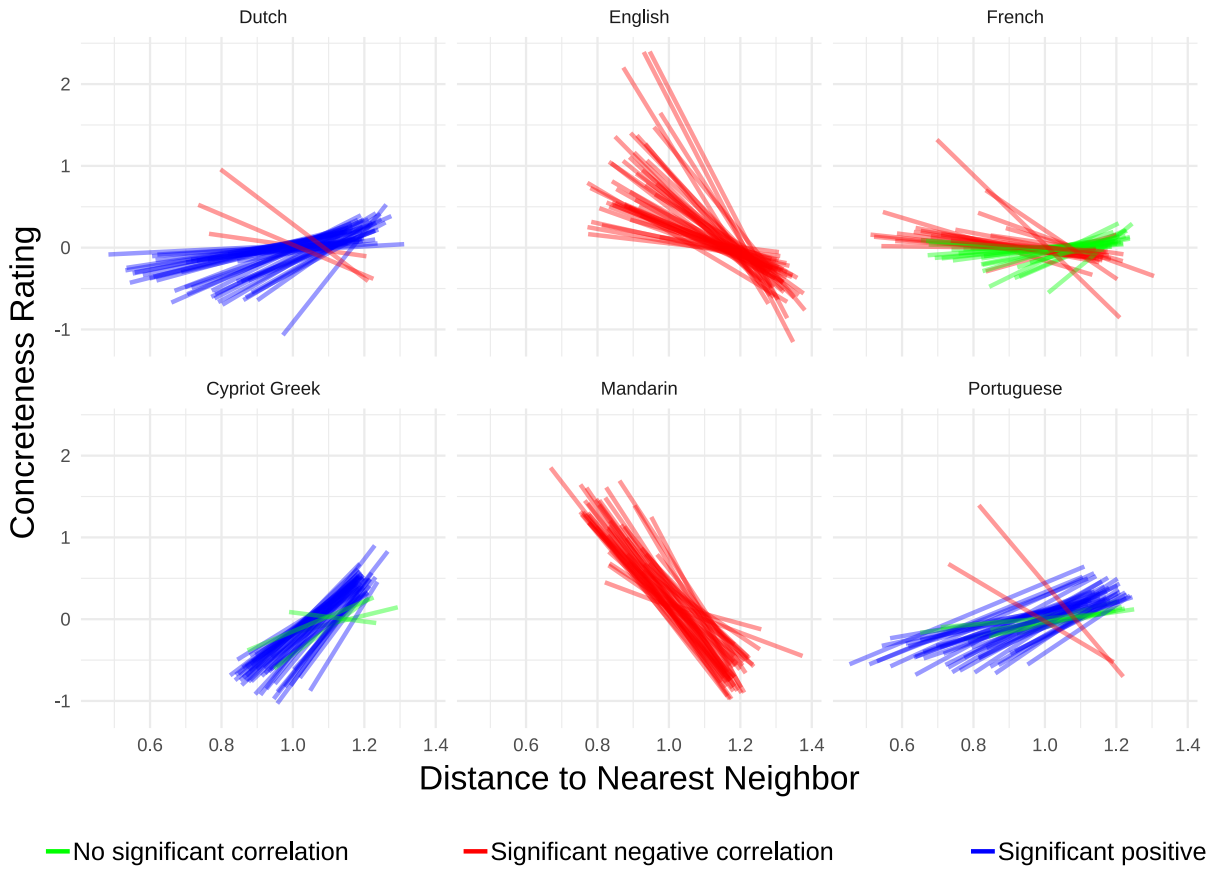


Figure 1: Relationship between normalized concreteness ratings (y axis) and distance to nearest cross-linguistic neighbor (x axis) across six rating languages (panels). Lines show simple linear regressions (concreteness ∼ distance), with color indicating the direction and significance of the distance effect in the multiple regression controlling for word frequency.

4 Discussion

Our results suggest a lack of a uniform relationship between cross-linguistic semantic relatedness and concreteness ratings across languages. English, Mandarin, and part of the French data are consistent with the hypothesis that concrete words are more similar cross-linguistically. However, Dutch, Portuguese, and Cypriot Greek showed the opposite pattern, suggesting that concrete words may be more language-specific than less concrete words.

One possible interpretation is that speakers of different languages associate concreteness with different semantic properties, despite the similar instructions used to elicit the ratings across languages. To examine this possibility more closely, we could identify overlapping concepts from different ratings datasets and check if their ratings are positively correlated. Another possible source of variation is the lexical composition of the ratings datasets, as some cover a broader range of syntactic categories, including highly abstract function words, whereas some do not, resulting in discrepancies among the effective endpoints of the rating scale. However, this explanation cannot fully account for the contrasting patterns of English and Dutch, whose datasets are comparatively similar in size and lexical coverage. Further investigation is therefore needed to determine which properties of the datasets or languages give rise to these differences. More broadly, the present findings highlight the importance of examining what "concreteness" captures for speakers of different languages.

While prior work has demonstrated that distributional semantic representations can be used to predict concreteness ratings (Thompson and Lupyan, 2018; Charbonnier and Wartena, 2020), the present study takes a more explanatory approach. Rather than treating embeddings as a predictive tool based on existing concreteness norms, we use them to probe what concreteness captures across languages and to what extent it reflects shared semantic or-

ganization. Through this attempt, we demonstrate how word embeddings can be used to operationalize psycholinguistic constructs involving cross-linguistic semantic relatedness, providing a quantitative complement to traditional ratings-based measures.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Patrick Bonin, Alain Méot, and Aurélia Bugaiska. 2018. Concreteness norms for 1,659 French words: Relationships with other psycholinguistic variables and word recognition times. *Behavior Research Methods*, 50(6):2366–2387.

Sarah D Breedin, Eleanor M Saffran, and H Branch Coslett. 1994. Reversal of the concreteness effect in a patient with semantic dementia. *Cognitive Neuropsychology*, 11(6):617–660.

Marc Brysbaert and Boris New. 2009. Moving beyond Kučera and Francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English. *Behavior Research Methods*, 41(4):977–990.

Marc Brysbaert, Michaël Stevens, Simon De Deyne, Wouter Voorspoels, and Gert Storms. 2014a. Norms of age of acquisition and concreteness for 30,000 Dutch words. *Acta Psychologica*, 150:80–84.

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

Qing Cai and Marc Brysbaert. 2010. SUBTLEX-CH: Chinese word and character frequencies based on film subtitles. *PloS One*, 5(6):e10729.

Jean Charbonnier and Christian Wartena. 2020. Predicting the concreteness of german words. In *SWISSTEXT & KONVENS 2020: Swiss Text Analytics Conference & Conference on Natural Language Processing 2020; Proceedings of the 5th Swiss Text Analytics Conference (SwissText) & 16th Conference on Natural Language Processing (KONVENS), CEUR Workshop Proceedings Vol. 2624*.

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

Aloysia M Gorman. 1961. Recognition memory for nouns as a function of abstractness and frequency. *Journal of Experimental Psychology*, 61(1):23.

Armand Joulin, Piotr Bojanowski, Tomáš Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2984.

Emmanuel Keuleers, Marc Brysbaert, and Boris New. 2010. SUBTLEX-NL: A new measure for Dutch word frequency based on film subtitles. *Behavior Research Methods*, 42(3):643–650.

Judith F Kroll, Susan C Bobb, and Zofia Wodniecka. 2006. Language selectivity is the exception, not the rule: Arguments against a fixed locus of language selection in bilingual speech. *Bilingualism: Language and Cognition*, 9(2):119–135.

Wallace E Lambert. 1955. Associational fluency as a function of stimulus abstractness. *Canadian Journal of Psychology/Revue Canadienne de Psychologie*, 9(2):103.

Guido Löhr. 2021. What are abstract concepts? on lexical ambiguity and concreteness ratings. *Review of Philosophy and Psychology*, 13:549—-566.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Mark Myslín and Roger Levy. 2015. Code-switching and predictability of meaning in discourse. *Language*, 91(4):871–905.

Allan Paivio. 1963. Learning of adjective-noun paired associates as a function of adjective-noun word order and noun abstractness. *Canadian Journal of Psychology/Revue Canadienne de Psychologie*, 17(4):370.

Allan Paivio. 1966. Latency of verbal associations and imagery to noun stimuli as a function of abstractness and generality. *Canadian Journal of Psychology/Revue Canadienne de Psychologie*, 20(4):378.

Allan Paivio. 1986. *Mental Representations: A Dual Coding Approach*, chapter 4. Oxford University Press, New York.

John TE Richardson. 1975. Concreteness and imageability. *The Quarterly Journal of Experimental Psychology*, 27(2):235–249.

Paula J Schwanenflugel and Carolyn E Akin. 1994. Developmental trends in lexical decisions for abstract and concrete words. *Reading Research Quarterly*, (3):251–264.

Paula J Schwanenflugel, Katherine Kip Harnishfeger, and Randall W Stowe. 1988. Context availability and lexical decisions for abstract and concrete words. *Journal of Memory and Language*, 27(5):499–520.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax.

Ana Paula Soares, Ana Santos Costa, João Machado, Montserrat Comesaña, and Helena Mendes Oliveira. 2017. The Minho word pool: Norms for imageability, concreteness, and subjective frequency for 3,800 Portuguese words. *Behavior Research Methods*, 49(3):1065–1081.

Robyn Speer. 2022. rspeer/wordfreq: v3.0.

Insup Taylor. 1976. Similarity between French and English words—a factor to be considered in bilingual language behavior? *Journal of Psycholinguistic Research*, 5(1):85–94.

Bill Thompson and Gary Lupyan. 2018. Automatic estimation of lexical concreteness in 77 languages. In *Proceedings of the annual meeting of the cognitive science society*, volume 40, pages 1120–1125.

Artemis Traikapi, Michaela Michael, and Nikos Konstantinou. 2024. The Greek-Cypriot word pool: Normative data for concreteness and imageability for 2,950 words.

Janet van Hell and Annette MB de Groot. 1998a. Disentangling context availability and concreteness in lexical decision and word translation. *The Quarterly Journal of Experimental Psychology: Section A*, 51(1):41–63.

Janet G van Hell and Annette MB de Groot. 1998b. Conceptual representation in bilingual memory: Effects of concreteness and cognate status in word association. *Bilingualism: Language and Cognition*, 1(3):193–211.

Xu Xu and Jiayin Li. 2020. Concreteness/abstractness ratings for two-character Chinese words in MELD-SCH. *PloS One*, 15(6):e0232133.

Frequency modulates structural choice in Turkish suspended affixation: a latent-process account

Utku Turk[1], Eva Neu[2], Ozge Bakay[2], Brian Dillon[2], Gaja Jarosz[2]

[1]University of Maryland, College Park
[2]University of Massachusetts, Amherst
Correspondence: utkuturk@umd.edu

Abstract

Suspended affixation (SA) allows a suffix on one conjunct to scope over all coordinated elements. While inflectional SA is productive in Turkish, derivational SA is claimed to be highly restricted; yet speakers readily accept certain cases. We propose that this gradient acceptability reflects a frequency-modulated choice between two possible syntactic representations: base-generation, which licenses derivational SA, and ellipsis. To test this, we conducted a rating task on the acceptability of four derivational suffixes in SA form while manipulating the frequency of coordinations. Using a Multinomial Processing Tree model to isolate latent structural choices from surface ratings, we found that frequency modulated SA acceptability for some suffixes (i.e., *-sIz* '-less' and *-cI* '-maker'), but not others (i.e., *-lI* '-having' and *-lIk* '-for'). These findings suggest that frequency shapes syntactic parsing in morphologically complex environments.

1 Introduction

A central puzzle in the morphosyntax of agglutinative languages concerns the distribution of affixes across coordinated structures. In Turkish, a single suffix appearing on the rightmost conjunct can be semantically interpreted as scoping over multiple conjuncts—a phenomenon termed *suspended affixation* (SA). Consider (1), where the dative marker *-ye* surfaces only on the second conjunct yet receives an interpretation in which both *Ali* and *Ayşe* bear dative case.

(1) Ali ve Ayşe-ye bak-tı-m.
Ali and Ayşe-DAT look-PST-1SG
'I looked at Ali and Ayşe.'

This construction has been documented across typologically diverse languages, including Turkish (Kornfilt, 1996, 2012; Kabak, 2007; Broadwell, 2008; Akkuş, 2016; Atmaca, 2021), Mari (Guseva and Weisser, 2018), Ossetic, Iron, and Eastern Armenian (Erschler, 2012), Dagur (Gong, 2021), Japanese and Korean (Yoon and Lee, 2005), Nivkh (Gruzdeva, 1998), and Hungarian (Trommer, 2008).

While SA with *inflectional* morphology is relatively uncontroversial, the availability of *derivational* SA remains theoretically contested. While some suggest that derivational SA is not a productive process in Turkish (Atmaca, 2021; Kabak, 2007; Kornfilt, 2012), others argue that it is more productive than previously assumed (Akkuş, 2016). Drawing on these conflicting observations, the current study empirically examines whether the gradient acceptability of derivational SA in Turkish is modulated by the frequency of coordinations. By fitting a Multinomial Processing Tree model to rating data on SA forms across different suffixes and coordinations with varying frequencies, we show evidence that frequency does not simply increase surface acceptability of all suffixes, but instead influences syntactic parsing for certain suffixes.

1.1 The status of derivational SA in Turkish

Earlier observations regarding derivational SA in Turkish report that it is a rare phenomenon with considerable inter-speaker and inter-item variation. For example, Kabak (2007) argues that apparent cases of derivational SA, such as (2), are restricted to tightly bound, lexicalized collocations (see also Atmaca, 2020).

(2) a. ana ve baba-lık
mother and father-DER
'motherhood and fatherhood'

b. ay-yıldız-lı bayrak
moon-star-ADJ flag
'a flag with a moon and star' (lit. 'a moon-star-ADJ flag', i.e., the Turkish flag)

Under his view, examples such as (2) are idiomatic

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 343–352
July 3-4, 2026 ©2026 Association for Computational Linguistics

chunks stored holistically in the lexicon and denoting frequently co-occurring entities (Wälchli, 2005), rather than reflecting a productive affixation process. Similarly, Kornfilt (2012) has argued that SA in Turkish is strictly limited to inflectional affixes.

However, subsequent work by Akkuş (2016) challenges this categorical restriction. Based on examples sourced online, Akkuş demonstrates that derivational SA across both nominal and verbal domains is more productive than previously thought. Crucially, he shows that the distributing affix can scope over full phrasal conjuncts—not merely adjacent, tightly-bound lexical items—e.g., '[5 lira] and [10 dollar]-DER' in (3).

(3) ... beş lira ve on dolar-lık
five lira and ten dollar-DER
banknot-lar...
banknote-PL
'5-lira and 10-dollar banknotes'

Moreover, some cases of derivational SA cited by Akkuş are acceptable regardless of the order of conjuncts, shown for the conjunct *dost ve arkadaş* 'buddy and friend' along with its version with a reversed order in (4). This flexibility is hard to reconcile with the idea that these SA constructions are simple isolated collocations stored in the lexicon.

(4) a. Bütün eğitim çalışmaları boyunca
all training sessions during
dost ve arkadaş-ça bir ortam
buddy and friend-DER an atmosphere
ol-malı-dır.
be-must-MOD
'There must be a friendly and intimate environment during the whole training session.'

b. ... buna uygun davran-mak
this-DAT appropriate behave-NMLZ
için arkadaş ve dost-ça
for friend and buddy-DER
gel-di-k.
come-1PL-PST
'... accordingly, we came in a friendly and intimate manner.'

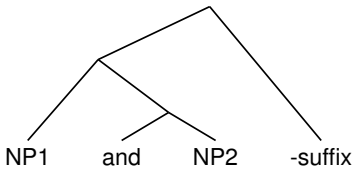
At the same time, the fact remains that derivational SA is more idiosyncratically restricted than inflectional SA. Specifically, we observe that it appears to be more readily accepted for high-frequency coordinations such as *moon and star* and *friend and buddy*, as correctly noted by Kornfilt and Kabak.

While Akkuş's findings have been adopted in subsequent work on SA (e.g., Atmaca, 2021; Türk, 2025), the role of frequency in modulating the availability and parse of these constructions has not been systematically investigated. Given the speaker- and item-level variation noted in the literature, we propose that the apparent conflict between the intuitions of Kabak (2007) and the evidence provided by Akkuş (2016) may be reconciled by attending to the *probabilistic* nature of syntactic choice underlying derivational SA.

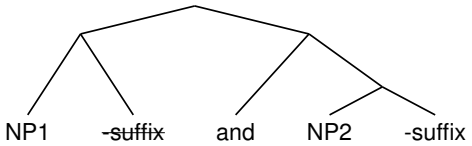
1.2 The present study

Our proposal in this study rests on the assumption that in general, SA is structurally ambiguous between two possible parses. Under a *base generation* analysis as in (5), the suspended affix combines with the coordinated noun phrase as a whole. Under an *ellipsis* analysis as in (6), the affix is underlyingly merged with each conjunct, but is deleted in the non-final conjunct(s). When encountering an SA construction, speakers are in principle free to adopt either parse.[1]

(5)



(6)



Against this background, we argue that the difference in productivity between derivational and inflectional SA can be accounted for as follows. First, we propose that derivational affixes can only participate in SA in a base generation structure and not in an ellipsis structure, whereas inflectional affixes can freely undergo ellipsis. What distinguishes derivational from inflectional affixes is that they sit closer to the root and are more tightly integrated with it semantically, phonologically and prosodically. We suggest that as a result, derivational affixes cannot be deleted at the exclusion of the root since the two form too inseparable a

[1] This holds for the items in our experimental studies. Some instances of SA are only compatible with one parse due to various syntactic, morphological and semantic factors. In Section 4, we discuss one such factor—scope interpretation—and how it can be used to further test our hypothesis that the gradient acceptability of derivational SA is linked to structural ambiguity.

unit. To our knowledge, this claim is novel. In Section 4, we discuss more in detail why it would apply; here, we focus on how it derives the gradient acceptability profile of derivational SA.

Second, we argue that the availability of the two structural paths to SA in (5) and (6) is shaped by the frequency of the coordinated structure. There is strong evidence that speakers store holistic representations of coordinated noun phrases, and that the more frequent the coordination is, the more likely speakers are to access this holistic representation in parsing (Morgan and Levy, 2015, 2016). Crucially, such a holistic representation is only compatible with a base generation analysis; in the ellipsis structure, the coordinated phrase such as *mother and father* does not form a constituent. Consequently, we propose that high-frequency structures bias the parser towards a base generation analysis.

Taken together, these two proposals account for the differences in productivity of inflectional and derivational SA. Since derivational SA is only available with a base generation analysis, it is more acceptable with high-frequency coordinations, as has been observed anecdotally in the literature and as we confirm experimentally in the following. Upon encountering a high-frequency coordinated phrase, speakers tend to access the holistic representation associated with it. This leads them to posit a base generation analysis, which is compatible with derivational SA. In contrast, inflectional SA is highly productive overall (Atmaca 2020, but see Kornfilt 1996, Türk 2025). This is as expected if inflectional affixes can freely participate in SA under both structural analyses. Even if speakers posit an ellipsis analysis of SA for low-frequency coordinations, the result is still acceptable.

Our claim that the frequency of coordinated noun phrases modulates structural choice in SA is rooted in much previous work showing evidence that frequency distributions can influence online parsing. For example, a large body of evidence supports the claim that frequency affects the processing of both single words and complex multi-word expressions (Pinker and Ullman, 2002; Arnon and Snider, 2010; Carroll and Conklin, 2020; Siyanova-Chanturia et al., 2011; Sosa and MacFarlane, 2002; Swinney and Cutler, 1979; Tremblay et al., 2011; Morgan and Levy, 2015, 2016). Frequent multi-word sequences are recognized faster and processed more fluently than matched infrequent sequences, even when individual word frequency is controlled. This suggests that speakers maintain holistic representations for recurrent phrases alongside compositional parsing.

Morgan and Levy (2015, 2016) show evidence that frequency-based storage interacts with other grammatical preferences. They show that more frequent coordinated phrases exhibit more rigid ordering preferences. For example, *ladies and gentlemen* occurs almost exclusively in this order, whereas *towels and sunglasses* alternates freely with *sunglasses and towels*. Morgan and Levy interpret this as a trade-off between compositional and holistic representations: as a coordination becomes more frequent, speakers increasingly store it as a unit rather than assembling it compositionally, resulting in frozen ordering.

A key finding that is relevant for the present study is that the effect of frequency does not have to be linear. Alternatively, the effect of frequency can be seen as a choice between competing syntactic structures, as has been suggested for the domain of argument structure alternations, most notably the English dative alternation. Bresnan and colleagues (Bresnan, 2007; Bresnan and Ford, 2010; Bresnan et al., 2004) have shown that the choice between a double object construction (*I gave Mary the book*) and a prepositional dative (*I gave the book to Mary*) is probabilistic rather than categorical, and is governed by a range of factors including animacy, givenness, and constituents' phonological weight. Crucially, verb bias—the frequency with which a verb appears in one construction over the other—shapes expectations about structural form and influences both production and comprehension, with speakers' naturalness ratings closely tracking corpus-derived construction probabilities (Bresnan, 2007).

Similarly, our proposal goes beyond the claim that frequent expressions are simply stored as unanalyzed wholes. While we draw on Morgan and Levy's idea that higher frequency coordinations are more likely to be represented holistically, our critical prediction concerns not storage *per se*, but its downstream consequences for syntactic structure assignment. In our model, the key variable modulated by frequency is the probability of adopting a base-generation parse over an ellipsis parse. Highly frequent coordinations are more likely to be accessed holistically, making the base-generation structure—in which the affix attaches to the coordination as a single constituent—more readily available. Lower-frequency coordinations remain compatible with both routes, but the ellipsis

parse—in which the affix merges with each conjunct separately—becomes relatively more available.

Thus, while we see an effect of frequency in the experimental results, we show that frequency does not uniformly boost acceptability scores across the board. This would not be an informative finding, but might simply indicate that speakers prefer expressions that they have encountered more often in their linguistic input. Instead, we argue that frequency shifts the distribution over latent structural analyses, and these structural analyses in turn determine whether a given affix type can participate in SA. More specifically, we present a formal hypothesis that captures how frequency and structural ambiguity interact to yield the gradient acceptability profile observed in Turkish derivational SA using Multinomial Processing Trees (Riefer and Batchelder, 1988; Erdfelder et al., 2009), which represent the observed acceptability responses as a process generated by a sequence of latent subprocesses rather than by a single undifferentiated acceptance probability (Riefer and Batchelder, 1988; Batchelder and Riefer, 1999; Klauer, 2010). Our model rests on the assumptions that the structural choice between a base generation and an ellipsis analysis is probabilistic and sensitive to frequency, and furthermore, that participants are not always attentive in their responses.

We proceed as follows. Section 2 presents experimental data confirming a frequency effect for SA with certain kinds of derivational affixes. Section 3 then introduces a Multinomial Processing Tree model of the results. We end on a general discussion in Section 4.

2 Experiment

We conducted a Likert-scale judgment task to test whether the acceptability of SA involving derivational suffixes is modulated by the frequency of the coordinated phrases. We predicted that if derivational SA is sensitive to frequency, the acceptability of coordinated phrases with SA would increase as a function of their frequency. Furthermore, if this sensitivity is a general property, we should observe comparable frequency effects across all tested suffixes.

2.1 Materials and methods

The stimuli were Turkish sentences including a coordinated phrase as a modifier of a noun phrase in the sentence. We employed a 2x4 within-subject design, crossing SA, i.e., coordinated phrases with SA (i.e., with the derivational suffix appearing only on the second conjunct) or in FULL form (i.e, with the derivational suffix appearing on both conjuncts), and SUFFIX TYPE, i.e., *-cI* '*-maker*', *-lI* '*-having*', *-lIk* '*-for*' or *-sIz* '*-less*'. All four suffixes are highly productive in forming modifiers in the language. Example stimuli are in Table 1. The eight experimental conditions were distributed across eight lists using a Latin square design, ensuring that each participant saw each item only once and in only one condition.

There were 32 experimental sentences in total, each with a different coordinated noun phrase. The coordinated phrases covered a wide range of frequency: their counts were taken from a Turkish Web 2012 corpus (∼3.4 billion words), and ranged from 131,657 ('mother and father') to 1 ('mouse and lion'). To ensure semantic naturalness, coordinated nouns were taken from the same conceptual realm.

The experimental sentences were interspersed with 64 filler items. 24 of the filler items were ungrammatical due to several syntactic violations such as voice, case or number mismatches. The remaining 40 were grammatical sentences, 10 of which had coordinating adjectives with no affix, eliminating any ambiguity in syntactic parse, as in *red and thick scarf*.

47 Turkish speakers participated in the task. Their task was to judge the acceptability of sentences on a 6-point Likert scale (6 = 'very good'). An even-numbered scale was chosen to encourage participants to lean towards either an acceptable or unacceptable interpretation (Schütze and Sprouse, 2013). The task was conducted online on PCIbex-Farm (Zehr and Schwarz, 2018).

2.2 Results

Figure 1 shows the observed mean ratings as a function of centered rank frequency, with higher values corresponding to more frequent coordinations. Descriptively, neither the SA penalty nor the effect of frequency seems to be uniform across suffixes.

To analyze the ordinal rating data, we fit separate Bayesian cumulative-logit models for each suffix using `brms` (Bürkner, 2017). The models included fixed effects for SA, centered rank frequency, and their interaction, as well as varying effects for subjects and items. SA was sum contrast coded, with .5 for FULL and -.5 for SA forms. We

SUFFIX	Sentence								
-cI '-maker'	Selin	tamir	için	kapı(-cı)	ve	pencere-ci	bir	marangozla	anlaşmış.
	S.	repair	for	door-MAKER	and	window-MAKER	a	carpenter	made.a.deal.with
	'Selin agreed with a door-maker and window-maker carpenter for the repair.'								
-lI '-having'	Selin	tamir	için	kapı(-lı)	ve	pencere-li	bir	atölye	hazırlamış.
	S.	repair	for	door-HAVING	and	window-HAVING	an	atelier	prepared
	'Selin prepared an atelier with a door and a window for the repair.'								
-lIk '-for'	Selin	tamir	için	kapı(-lık)	ve	pencere-lik	bir	desen	seçmiş.
	S.	repair	for	door-FOR	and	window-FOR	a	pattern	chose
	Selin chose a pattern for a door and a window for the repair.								
-sIz '-less'	Selin	tamir	için	kapı(-sız)	ve	pencere-siz	bir	atölye	hazırlamış.
	S.	repair	for	door-LESS	and	window-LESS	an	atelier	prepared
	'Selin prepared an atelier without a door or a window for the repair.'								

Table 1: Example stimuli from the acceptability rating task. Suffixes in parentheses are omitted in forms with SA.

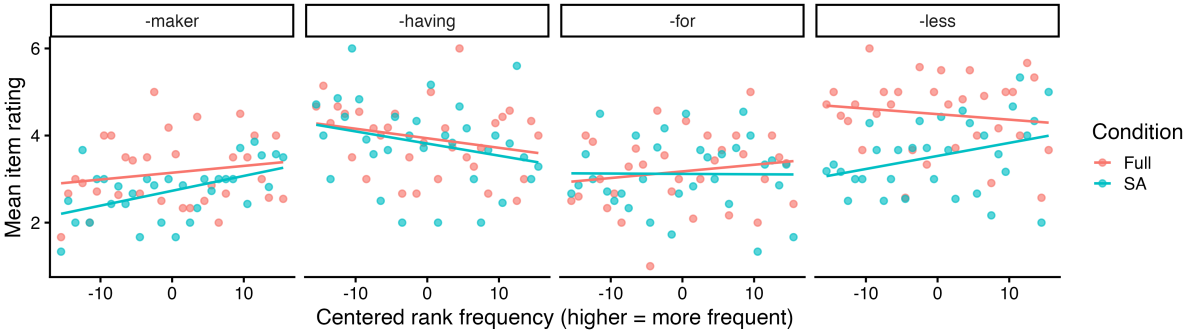


Figure 1: Observed mean item ratings by suffix and condition. Higher centered-rank values correspond to more frequent coordinations.

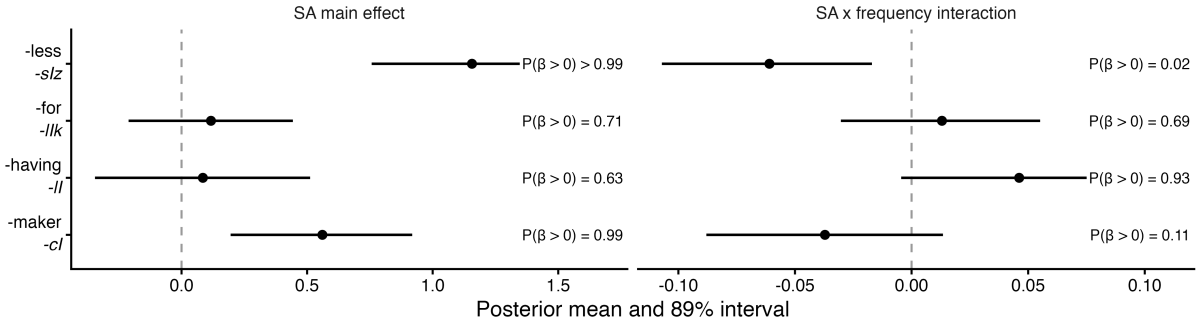


Figure 2: Posterior summaries from the suffix-specific Bayesian ordinal models. The left facet shows the SA main effect; the right facet shows the SA-by-frequency interaction. Labels at the right edge give $P(\beta > 0)$ for each coefficient.

also examined the potential role of trial order. Effects were mostly negligible for three of the four suffix-specific models (*-sIz*: $\hat{\beta} = 0.00$, 89% CrI $[-0.16, 0.17]$, $P(\beta{>}0) = 0.51$; *-cI*: $\hat{\beta} = 0.01$, 89% CrI $[-0.16, 0.17]$, $P(\beta{>}0) = 0.53$; *-lIk*: $\hat{\beta} = -0.03$, 89% CrI $[-0.19, 0.13]$, $P(\beta{>}0) = 0.38$). However, we found a negative main effect for *-lI* ($\hat{\beta} = -0.21$, 89% CrI $[-0.38, -0.05]$, $P(\beta{>}0) = 0.02$). Since it did not have a reliable effect on the ratings of all suffixes, it was excluded from the models reported here.

Figure 2 plots the posterior means, their 89% intervals and the posterior probability of a positive effect for SA and its interaction with frequency. In Figure 2, a positive sign for the SA effect suggests higher ratings for FULL relative to SA forms; a negative sign for the interaction effect suggests that frequency improves the ratings for SA forms more than it improves those for their FULL versions.

Results showed an overall decrease in the ratings

for SA relative to FULL forms with '-less' ($\hat{\beta}$=1.16, 89%CrI [0.76, 1.55]) and '-maker' ($\hat{\beta}$=0.56 [0.20, 0.92]). There was no such difference with '-having' or '-for'. The clearest evidence for an interaction between form and frequency was with '-less', evidenced by an increase in the acceptability of SA forms with higher frequency ($\hat{\beta}$=−0.06 [-0.11, -0.02]). There was no clear evidence for an interaction with '-maker' or '-for'. Lastly, '-having' showed a trend in the opposite direction, suggesting a decrease in the acceptability of SA forms with higher frequency; but this trend was not statistically reliable ($\hat{\beta}$=0.05 [0.00, 0.10]).

Overall, there was clear evidence that the four suffixes tested here differ with respect to the overall SA penalty and how this penalty is modulated by frequency. This suggests that a simple narrative where frequency uniformly boosts acceptability is insufficient. Moreover, the cumulative-logit models we report above can only describe final ratings, which entangle structural parsing preferences with baseline response biases and inattention. To systematically reveal the latent variables contributing to the ratings and determine the exact role of frequency in syntactic choice, we use a Multinomial Processing Tree process model, which we discuss next.

3 Latent Modeling

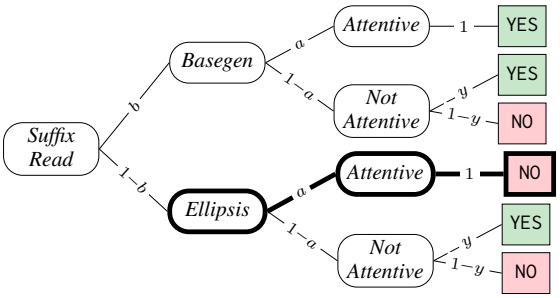
3.1 Methods

It is widely accepted that experimental measures such as acceptability judgments or reading times are not transparent reflections of a single underlying process; rather, they are generated by a mixture of latent subprocesses (Ratcliff, 1978; Riefer and Batchelder, 1988; Batchelder and Riefer, 1999; Paape and Vasishth, 2022; Nicenboim et al., 2024). Consequently, inferences based only on condition means can underrepresent some cognitive processes while amplifying others. On this view, assuming a simple, one-to-one mapping between frequency and acceptability is too coarse: frequency may influence ratings indirectly by shifting structural choice, while the observed responses are simultaneously shaped by attention and response bias. The descriptive results above align with this point, because neither frequency nor the availability of derivational SA exerts a single linear effect across suffixes, suggesting a potential role for subprocesses in acceptance ratings.

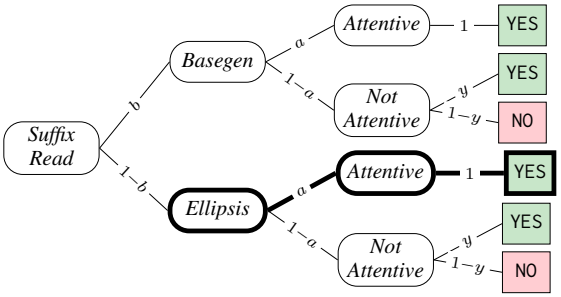
Therefore, we analyze the experimental findings using Multinomial Processing Trees (Riefer and Batchelder, 1988; Erdfelder et al., 2009; Batchelder and Riefer, 1999; Klauer, 2010). We make three main assumptions in the model. First, the structural choice between a base-generation and an ellipsis analysis is probabilistic rather than deterministic, and *probabilities for structural choices* can shift as a function of experience-based factors such as frequency (Jurafsky, 1996). The second assumption is that participants may not always be attentive in their responses. When they are inattentive, their responses are not random; instead, they are influenced by an a priori response yes/no tendency, which we model as a *subject-level response-bias*. Third, in addition to differences across participants, a single participant's attentional state may vary from trial to trial. While a participant may sometimes respond on the basis of the stimulus, they may sometimes fail to do so (Paape and Vasishth, 2022; Nicenboim et al., 2024). We model this as *trial-wise attentional variation*.

Accordingly, our model includes three latent subprocesses: structural choice (base generation vs. ellipsis), attentional state (i.e., when participants are not attending to the stimulus), and a residual yes-response tendency in an inattentive state. The trees in (7) and (8) illustrate the logic schematically:

(7) Derivational Suffix



(8) Inflectional Suffix



In the derivational case, attentive processing leads to acceptance under the base-generated parse but not under the ellipsis parse; in the inflectional case, both structural routes remain compatible with acceptance. In both trees, inattentive trials fall back

to a participant-specific yes-bias. We predict that if frequency modulates structural choice in parsing derivational SA, we should observe a higher probability for a base generation route with more frequent coordinations.

Before fitting the MPT, we converted 6-point ratings to binary accept/reject outcomes. The threshold was chosen empirically from filler items by fitting a Bayesian cumulative-logit model and evaluating each candidate threshold by its posterior balanced accuracy in separating grammatical from ungrammatical fillers. A threshold at 4 gave the best performance (balanced accuracy = 0.929, 90% interval: [0.914, 0.942]); ratings of 4–6 were treated as acceptance and 1–3 as rejection.

The MPT was then fit to these binary judgments. Let $a_{j[n]}$ denote subject j's attentiveness on trial n and $g_{j[n]}$ their filler-derived yes-bias. For SA items, the attentive state involves a further latent choice: with probability b_n the parser adopts a base-generation analysis and the SA item inherits full-form acceptance θ_n^{full}; with probability $1 - b_n$ it adopts an ellipsis analysis governed by θ_n^{ell}:

$$(9) \quad P(\text{yes}_n) = a_{j[n]} \left[b_n \, \theta_n^{\text{full}} + (1 - b_n) \, \theta_n^{\text{ell}} \right] + (1 - a_{j[n]}) \, g_{j[n]}$$

For FULL-form items $b_n = 1$ by assumption, which simplifies Equation (9) to (10):

$$(10) \quad P(\text{yes}_n) = a_{j[n]} \, \theta_n^{\text{full}} + (1 - a_{j[n]}) \, g_{j[n]}$$

The latent subprocesses of the MPT model shown in (7) and (8) are as follows.

Attention (a_j): Each participant's attentiveness probability, governing whether their response reflects the structural model or falls back to their default yes-bias, calculated by their responses to grammatical fillers.

Criterion-derived yes bias (g_j): Estimated from each participant's hits and false alarms on filler items via a signal-detection-theoretic criterion c_j, mapped to a yes-bias through $\Phi(-c_j)$. Using exclusively fillers ensures that response bias and attentiveness are identified independently of the critical SA items (Hautus et al., 2021).

Attentive acceptance (θ_n^{full}): Suffix-specific acceptance probability for FULL forms, with varying effects for subjects and items.

Structural route (b_n): For SA forms, the probability of adopting a base-generation analysis (Equation (9)). Under base generation, the SA item inherits θ_n^{full}; under ellipsis $(1 - b_n)$, acceptance is governed by a separate suffix-specific parameter θ_n^{ell}.

Frequency effect on b_n: The base-generation probability is modeled on the logit scale with a suffix-specific intercept and slope for centered rank frequency, plus varying effects for subjects and items. A positive slope means higher frequency favors base generation—the structural route that licenses derivational SA.

The MPT model was implemented directly in Stan (Carpenter et al., 2017).

3.2 Results

The MPT outperformed a maximal `brms` baseline on PSIS-LOO (ΔELPD = 7.47; Sivula et al. 2025; Vehtari et al. 2017), and is therefore used as the primary model. Nuisance parameters were well-identified: mean attentiveness was 0.81 ($SD = 0.01$), and the filler-derived criterion was mildly liberal ($M = -0.24$, $SD = 0.41$), corresponding to an inattentive yes-bias of 0.59 ($SD = 0.15$), ensuring that structural parameters do not absorb response bias.

Table 2 reports the structural parameters. SA-form acceptability (row 2) is not a free parameter: it is the weighted average $b \cdot \theta_{\text{full}} + (1 - b) \cdot \theta_{\text{ellip}}$, so the gap between rows 1 and 2 reflects both how penalized the ellipsis route is and how often it is taken.

Two patterns stand out. First, the SA penalty varies substantially across suffixes: largest for '-less' and '-maker', modest for '-having', and negligible for '-for'. The '-for' case shows why frequency cannot act as a direct acceptability booster: when both routes already yield poor acceptance, shifting route probability has little observable effect on ratings.

Second—and this is the central contribution of the MPT—frequency shifts $P(\text{base-gen})$ in the predicted direction for the two suffixes with the clearest SA penalty ($P(\beta{>}0) = 0.83$ for '-less'; 0.80 for '-maker'). The '-maker' effect had no surface signature in the ordinal model, because the low full-form acceptability (0.34) limits the observable gain from route-shifting. For '-having', the frequency effect runs negative ($P(\beta{>}0) = 0.12$), and for '-for', it is uninformative ($P(\beta{>}0) = 0.62$).

Overall, frequency is better understood as modulating structural route availability than as adding a uniform boost to surface acceptability.

Parameter	Suffix	M [Lower, Upper]	Interpretation
Full-form acceptability	*-sIz* '-less'	0.86 [0.79, 0.92]	Attentive-state $P(\text{accept})$ for full forms
	-cI '-maker'	0.34 [0.24, 0.44]	
	-lI '-having'	0.77 [0.69, 0.85]	
	-lIk '-for'	0.36 [0.26, 0.46]	
SA-form acceptability	*-sIz*	0.49 [0.37, 0.62]	Marginal $P(\text{accept})$ at mean frequency
	-cI	0.19 [0.11, 0.29]	
	-lI	0.65 [0.53, 0.76]	
	-lIk	0.32 [0.23, 0.42]	
Ellipsis acceptability	*-sIz*	0.33 [0.17, 0.52]	Attentive-state $P(\text{accept})$ via ellipsis route
	-cI	0.11 [0.04, 0.21]	
	-lI	0.52 [0.32, 0.70]	
	-lIk	0.24 [0.11, 0.38]	
$P(\text{base-gen})$ at mean freq.	*-sIz* ($P(\beta{>}0.5){=}0.08$)	0.30 [0.11, 0.54]	Ellipsis strongly dominates
	-cI ($P(\beta{>}0.5){=}0.26$)	0.38 [0.12, 0.71]	Ellipsis dominates
	-lI ($P(\beta{>}0.5){=}0.50$)	0.50 [0.19, 0.81]	Completely uncertain
	-lIk ($P(\beta{>}0.5){=}0.69$)	0.60 [0.23, 0.89]	Base-gen. modestly dominant
Effect of freq. on $P(\text{base-gen.})$	*-sIz* ($P(\beta{>}0){=}0.83$)	$+0.08$ [-0.06, $+0.26$]	suggestive positive effect
	-cI ($P(\beta{>}0){=}0.80$)	$+0.15$ [-0.16, $+0.45$]	suggestive positive effect
	-lI ($P(\beta{>}0){=}0.12$)	-0.20 [-0.49, $+0.08$]	suggestive negative effect
	-lIk ($P(\beta{>}0){=}0.62$)	$+0.06$ [-0.33, $+0.43$]	uncertain

Table 2: Posterior means and 89% credible intervals for key MPT parameters. In the lower section, parenthetical values next to each suffix are $P(\beta{>}\text{chance})$: the posterior probability that base generation is the dominant parse at mean frequency. In the frequency effect section, parenthetical values are $P(\beta{>}0)$: the posterior probability that the effect of frequency on base-generation probability is positive. The frequency effect is on the logit scale.

4 General Discussion

This study investigated how lexical frequency modulates the acceptability of derivational suspended affixation and modeled the underlying mechanism using a Multinomial Processing Tree. While our ordinal regression models demonstrate that suffixes do not behave uniformly with respect to frequency, the MPT's central contribution was to explicitly map out the underlying mechanism driving these differences. Specifically, the MPT model decomposes what might look like a direct frequency-acceptability link into a frequency effect on structural route probability: higher-frequency items are more likely to receive a base-generation analysis, and acceptability then follows from this structural choice together with nuisance processes. Critically, this decomposition revealed a frequency effect on structural choice for '-maker' that was entirely absent from the surface ordinal ratings—demonstrating that condition means alone are insufficient and that the computational model is necessary to isolate latent structural preferences.

Not all suffixes showed the predicted frequency-driven increase in base-generation probability, and the MPT explains why. For '-for', base generation is already the modestly dominant parse at mean frequency ($P(b{>}0.5) = 0.69$), and both full-form and SA-form acceptability are low—a configuration in which frequency-driven route-shifting yields little observable leverage on ratings. For '-having', frequency moved base-generation probability in the negative direction ($P(\beta{>}0) = 0.12$), possibly reflecting this suffix's relatively higher ellipsis-route acceptability compared to the other suffixes. These patterns are not unexplained residuals: they follow naturally from the model, because the observable consequence of route-shifting depends on the suffix-specific acceptability. It is also possible that these null or unexpected trends reflect a lack of statistical power to detect smaller effects due to our limited number of participants. Given the robust effects observed with the other suffixes, we consider this less likely, but leave further testing with a larger sample size to future work.

A crucial question raised by our proposal is why certain suffixes would resist an ellipsis analysis in the first place—which we have assumed rather than motivated—and what role is played by the difference between derivational and inflectional morphemes. Theoretical frameworks such as Distributed Morphology (Halle and Marantz, 1993, 1994) have argued that the traditional derivational/inflectional divide is better conceptualized as a continuum determined by syntactic attachment height and the position of functional heads in the syntactic spine. Suffixes that attach low in

the structure—close to the root—are more lexically integrated and therefore a poorer target for deletion. This predicts that suffixes like '-less' and '-maker', whose base-generation probability is most strongly modulated by frequency, are precisely those whose attachment height makes ellipsis less viable. Making this link precise by independently characterizing the attachment heights of Turkish derivational suffixes is an important direction for future work. More generally, connecting our proposal more closely to the syntactic literature on ellipsis analyses of SA (e.g., Despić, 2017; Erschler, 2018; Guseva and Weisser, 2018; Türk, 2025) would sharpen the predictions considerably.

We see an additional future direction for testing and extending the account. The base generation and ellipsis parses make different scopal predictions: under base generation, the derivational suffix takes scope over the full coordination, applying to both conjuncts jointly, while under ellipsis, each conjunct contains the suffix independently. These parses are therefore distinguishable by their interpretation. For example, *kitap ve dergi-siz* under a base-generation parse means 'lacking books-and-magazines (jointly)', whereas under ellipsis, the meaning is 'lacking books and lacking magazines (separately).' If frequency biases readers towards base generation as the MPT suggests, higher-frequency items should preferentially support the joint, wide-scope reading in a forced-choice interpretation task. This task would provide direct, independent evidence that frequency operates on parse selection rather than on acceptability directly–the core theoretical claim of this paper.

References

Faruk Akkuş. 2016. Suspended affixation with derivational suffixes and lexical integrity. In Proceedings of IMOG 2016, Patras.

Inbal Arnon and Neal Snider. 2010. More than words: Frequency effects for multi-word phrases. Journal of Memory and Language, 62:67–82.

Furkan Atmaca. 2020. Suspended Affixation in Turkish. MA Thesis, Boğaziçi University.

Furkan Atmaca. 2021. Suspended affixation needs no morphological word: The suffix -(y)Ip. In Proceedings of the Workshop on Turkic and Languages in Contact with Turkic, volume 6, pages 5035–5035.

William H. Batchelder and David M. Riefer. 1999. Theoretical and empirical review of multinomial process tree modeling. Psychonomic Bulletin & Review, 6:57–86.

Joan Bresnan. 2007. Is syntactic knowledge probabilistic? Experiments with the English dative alternation. In Sam Featherston and Wolfgang Sternefeld, editors, Roots: Linguistics in Search of Its Evidential Base, pages 77–96. Mouton de Gruyter, Berlin.

Joan Bresnan, Anna Cueni, Tatiana Nikitina, and R. Harald Baayen. 2004. Predicting the dative alternation. In Gerlof Bouma, Irene Kraemer, and Joost Zwarts, editors, Cognitive Foundations of Interpretation, pages 69–94. Royal Netherlands Academy of Arts and Sciences, Amsterdam.

Joan Bresnan and Marilyn Ford. 2010. Predicting syntax: Processing dative constructions in American and Australian varieties of English. Language, 86:186–213.

George Aaron Broadwell. 2008. Turkish suspended affixation is lexical sharing. In Miriam Butt and Tracy Holloway King, editors, Proceedings of the LFG08 Conference, pages 198–213. CSLI Publications, Stanford.

Paul-Christian Bürkner. 2017. brms: An R package for Bayesian multilevel models using Stan. Journal of Statistical Software, 80:1–28.

Bob Carpenter, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus A. Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. 2017. Stan: A probabilistic programming language. Journal of Statistical Software, 76:1–32.

Gareth Carroll and Kathy Conklin. 2020. Is all formulaic language created equal? Unpacking the processing advantage for different types of formulaic sequences. Language and Speech, 63:95–122.

Miloje Despić. 2017. Suspended morphology in Serbian: Clitics vs. affixes. Glossa: a journal of general linguistics, 2:12.

Edgar Erdfelder, Tina-Sarah Auer, Benjamin E. Hilbig, Andreas Aßmann, and Morten Moshagen. 2009. Multinomial processing tree models: A review of the literature. Zeitschrift für Psychologie / Journal of Psychology, 217:108–124.

David Erschler. 2012. Suspended affixation in Ossetic and the structure of the syntax-morphology interface. Acta Linguistica Hungarica, 59:153–175.

David Erschler. 2018. Suspended affixation as morpheme ellipsis: Evidence from Ossetic alternative questions. Glossa: a journal of general linguistics, 3:12.

Zhiyu Mia Gong. 2021. Postsyntactic lowering and linear relations in Dagur noun phrases. Glossa: a journal of general linguistics, 6:42.

Ekaterina Gruzdeva. 1998. Nivkh. Languages of the World/Materials. Lincom Europa, Munich.

Elina Guseva and Philipp Weisser. 2018. Postsyntactic reordering in the Mari nominal domain: Evidence from suspended affixation. Natural Language & Linguistic Theory, 36:1087–1120.

Morris Halle and Alec Marantz. 1993. Distributed Morphology and the pieces of inflection. In Kenneth Hale and S. Jay Keyser, editors, The View from Building 20, pages 111–176. MIT Press, Cambridge, MA.

Morris Halle and Alec Marantz. 1994. Some key features of Distributed Morphology. In MIT Working Papers in Linguistics, pages 275–288.

Michael J Hautus, Neil A Macmillan, and C Douglas Creelman. 2021. Detection theory: A user's guide. Routledge.

Daniel Jurafsky. 1996. A probabilistic model of lexical and syntactic access and disambiguation. Cognitive Science, 20:137–194.

Barış Kabak. 2007. Turkish suspended affixation. Linguistics, 45:311–347.

Karl Christoph Klauer. 2010. Hierarchical multinomial processing tree models: A latent-trait approach. Psychometrika, 75:70–98.

Jaklin Kornfilt. 1996. On some copular clitics in Turkish. In ZAS Papers in Linguistics, volume 6, pages 96–114. Zentrum für Allgemeine Sprachwissenschaft, Berlin.

Jaklin Kornfilt. 2012. Revisiting "suspended affixation" and other coordinate mysteries. In Laura Brugè, Anna Ferro, Carla Ferraro, Anna Ferro, and Lisa Ferraro, editors, Functional Heads: The Cartography of Syntactic Structures, pages 181–196. Oxford UP, Oxford.

Emily Morgan and Roger Levy. 2015. Modeling idiosyncratic preferences: How generative knowledge and expression frequency jointly determine language structure. Proceedings of the 37th Annual Conference of the Cognitive Science Society, pages 1649–1654.

Emily Morgan and Roger Levy. 2016. Abstract knowledge versus direct experience in processing of binomial expressions. Cognition, 157:384–402.

Bruno Nicenboim, Daniel Schad, and Shravan Vasishth. 2024. An Introduction to Bayesian Data Analysis for Cognitive Science. Cambridge UP. Online version: https://bruno.nicenboim.me/bayescogsci/.

Dario Paape and Shravan Vasishth. 2022. Estimating the true cost of garden-pathing: A computational model of latent cognitive processes. Cognitive Science, 46:e13186.

Steven Pinker and Michael T. Ullman. 2002. The past and future of the past tense. Trends in Cognitive Sciences, 6:456–463.

Roger Ratcliff. 1978. A theory of memory retrieval. Psychological Review, 85:59.

David M. Riefer and William H. Batchelder. 1988. Multinomial modeling and the measurement of cognitive processes. Psychological Review, 95:318–339.

Carson T. Schütze and Jon Sprouse. 2013. Judgment data, pages 27–50. Cambridge University Press.

Tuomas Sivula, Måns Magnusson, Alejandro Andres Matamoros, and Aki Vehtari. 2025. Uncertainty in Bayesian leave-one-out cross-validation based model comparison. Bayesian Analysis.

Anna Siyanova-Chanturia, Kathy Conklin, and Walter J. B. van Heuven. 2011. Seeing a phrase "time and again" matters: The role of phrasal frequency in the processing of multiword sequences. Journal of Experimental Psychology: Learning, Memory, and Cognition, 37:776–784.

Anna Vogel Sosa and James MacFarlane. 2002. Evidence for frequency-based constituents in the mental lexicon: Collocations involving the word *of*. Brain and Language, 83:227–236.

David A. Swinney and Anne Cutler. 1979. The access and processing of idiomatic expressions. Journal of Verbal Learning and Verbal Behavior, 18:523–534.

Antoine Tremblay, Bruce Derwing, Gary Libben, and Chris Westbury. 2011. Processing advantages of lexical bundles: Evidence from self-paced reading and sentence recall tasks. Language Learning, 61:569–613.

Jochen Trommer. 2008. Hungarian has no portmanteau agreement. In Proceedings of the 26th West Coast Conference on Formal Linguistics, pages 478–486, Somerville, MA. Cascadilla Proceedings Project.

Utku Türk. 2025. Controlling morphosyntactic competition through phonology. In Pavel Caha, Karen De Clercq, and Guido Vanden Wyngaerd, editors, Nanosyntax and the Lexicalization Algorithm. Oxford UP, Oxford.

Aki Vehtari, Andrew Gelman, and Jonah Gabry. 2017. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. Statistics and Computing, 27:1413–1432.

Bernhard Wälchli. 2005. Co-Compounds and Natural Coordination. Oxford UP, New York, NY.

James Hye-Suk Yoon and Wooseung Lee. 2005. Conjunction reduction and its consequences for noun phrase morphosyntax in Korean. In Proceedings of the 24th West Coast Conference on Formal Linguistics, pages 379–387, Somerville, MA. Cascadilla Proceedings Project.

J. Zehr and F. Schwarz. 2018. PennController for Internet Based Experiments (IBEX).

Effect of case markers during agreement production: A model comparison using Armenian forced choice data

Pranab Bagartti, Samar Husain
Indian Institute of Technology Delhi, India
{pranab.bagartti, samar} @hss.iitd.ac.in

Abstract

Agreement attraction errors, where the verb erroneously agrees with a non-subject noun, have been a useful tool for investigating processes that subserve sentence production. Research has shown that case markers play an important role in modulating such errors. These effects have been argued to arise due to an underlying cue-based retrieval system. However, subsequent research in Armenian has challenged this conclusion (Avetisyan et al., 2020), arguing against a cue-based retrieval account.

The current paper revisits the Armenian production data through computational modeling. Specifically, we implemented three distinct models and compared their predictions; we compare (a) a cue-based retrieval model, (b) a feature migration model, and (c) a case as markers for agreement prediction model. Our model comparison results show that a case as markers for agreement prediction model followed by an inference component explains the effect of case better than the cue-based retrieval model as well as the feature migration model.

Keywords: agreement attraction, case marking, cue-based retrieval, feature migration, Armenian, sentence production, computational modeling

1 Introduction

Agreement attraction errors, where the verb erroneously agrees with a non-subject/non-target noun, have been used to investigate processes that subserve sentence production (e.g., Bock and Miller, 1991; Bock et al., 2001; Bock and Eberhard, 1993; Bock and Cutting, 1992; Eberhard et al., 2005; Badecker and Kuminiak, 2007; Vigliocco and Zilli, 1999; Eberhard et al., 2005; Eberhard, 1997; Staub, 2009, 2010; Franck et al., 2002). Research has shown that case markers play an important role in modulating such errors (e.g., Nicol and Anton-Mendez, 2009; Hartsuiker et al., 2001, 2003; Badecker and Kuminiak, 2007; Gillespie, 2011; Slioussar, 2018; Gillespie and Pearlmutter, 2013). For example, when speakers of Slovak were given incomplete sentences like (1a), where the distractor noun had an overtly distinctive case marker, they produced more accurate agreement at the verb in comparison to sentences like (1b), where the case on the distractor overtly resembled the subject's. Badecker and Kuminiak (2007) proposed that such errors arise due to an underlying cue-based retrieval system. Subsequent research in Armenian (Avetisyan et al., 2020), however, has challenged such an explanation, arguing against a cue-based retrieval account. The current paper utilizes model comparison to disambiguate between these diverging claims.

1. Example preambles from Slovak:

   a. Trest za
   punishment.M.**NOM** for
   vraždu...
   murder.F.**ACC**.SG...
   'The punishment for the murder...'

   b. Trest za
   punishment.M.**NOM** for
   krádež ...
   theft.F.**NOM/ACC**.SG ...
   'The punishment for the theft ...'

In this paper, we focus on the Armenian production data from Avetisyan et al. (2020) for two reasons. First, to the best of our knowledge, this is the only openly available production dataset on case-mediated agreement attraction with accuracy and reaction times. Second, Armenian is also a useful test case on linguistic grounds: a nominative case marker on a noun reliably co-occurs with subject-verb agreement within its clause, unlike in Slovak (Example 1), where a nominative noun can appear without triggering agreement. This tight coupling between nominative case and agreement makes it different from other languages investigated for case and agreement. Additionally, this coupling is also

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 353–364
July 3-4, 2026 ©2026 Association for Computational Linguistics

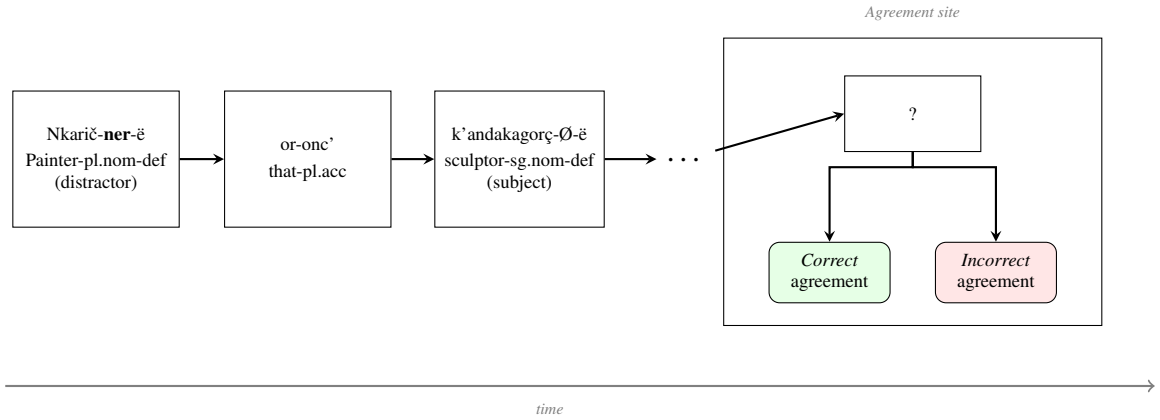


Figure 1: Schematic of the RSVP forced-choice task.

central to one of the theories that we posit moving forward (See §2.3).

In Avetisyan et al. (2020)'s task, sentences were presented one word at a time in a rapid manner (i.e., Rapid Serial Visual Presentation) followed by a forced choice between two continuations, i.e., correct or incorrect agreement (Staub, 2009, 2010). Importantly, the distractor appeared first, followed by the subject and then the agreement site where the participants choose from a correct and an incorrect response through key presses (See Figure 1).

The resulting data consisted of responses from 175 Eastern Armenian speakers. Avetisyan et al. (2020) found a limited/negligible effect of case with accuracy data but a more pronounced effect of case with their reaction time data. Does cue-based retrieval warrant such an effect?

A cue-based retrieval theory would assume that speakers use cues such as +nominative before producing an agreement. Any chunk that matches the cues gets a activation boost from the cue. In turn, the most activated chunk is retrieved. And finally, the verb shows agreement with the retrieved chunk.

Activation boost from the +nominative cue increases the chance of retrieving a subject noun when no other nouns match the cues as in 1a (also See Figure 2a). When a distractor matches the +nominative cue in sentences like 1b (also See Figure 2b), the activation boost is shared with the distractor, decreasing the chance of subject retrieval.

Regardless of the claim, whether a cue-based retrieval system can in fact produce the pattern of results reported by Avetisyan et al. (2020) is ultimately a question that can only be resolved through computational simulation. Such a computational modeling based evaluation has so far not been done. While cue-based retrieval has been computationally implemented for agreement attraction during comprehension (e.g., Lewis and Vasishth, 2005; Engelmann et al., 2019; Vasishth and Engelmann,

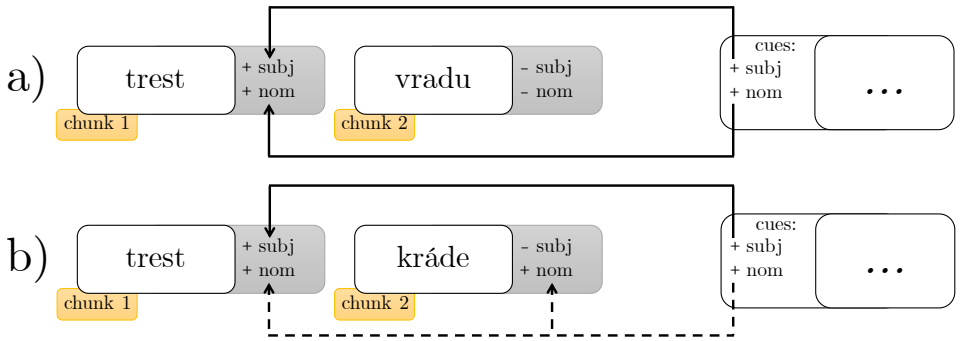


Figure 2: A schematic representation for agreement computation for 1a and 1b sentences from Slovak under a cue-based retrieval system. Here, +subj denotes a +subject feature/cue while +nom denotes a +nominative feature/cue. A +subject cue is consistent with previous implementations of cue-based retrieval for sentence processing (e.g., Engelmann et al., 2019; Yadav et al., 2022).

2021; Yadav et al., 2022), such an account remains unimplemented for agreement attraction during production.

In this paper, we present the first implementation of a cue-based retrieval model for such production data. In addition, implementing a single model in isolation has limited explanatory value. A good fit could reflect the flexibility of the model rather than the validity of its underlying mechanism. We therefore also implement other accounts proposed in Avetisyan et al. (2020), allowing us to evaluate which theoretical account best captures the observed pattern of agreement attraction in Armenian sentence production.

The rest of the paper is organised as follows. In the next section, we describe the different models under consideration. This is followed by a comparison between these models. We conclude with general discussion and limitations section.

2 Models of Subject Verb Agreement Production

The models we compare here are computational-level implementation of theoretical proposals about how the production system computes agreement; specifically, each model maps the experimental conditions in Avetisyan et al. (2020) onto a probability of correct agreement and a predicted reaction time. The free parameters of each model are estimated jointly from the accuracy and reaction time data using Hamiltonian Monte Carlo as implemented in Stan (Stan Development Team, 2026). Model comparison is done via leave-one-out cross-validation using the Pareto smoothed importance sampling estimator (Vehtari et al., 2024, 2017) in the `loo` package in R. This framework allows us to evaluate not just whether a given model can fit the data, but how well each model predicts held-out observations relative to the others.

All model priors were chosen to have similar predictions across models before looking at the data. Avetisyan et al. (2020)'s production experiment had 2×2 design with factors: case and number. Case factor had two levels, the first level had a case marked distractor while the second level had no case marking on the distractor. For the number factor, the distractor was either singular, matching with the subject's singular number, or plural, mismatching with the subject's singular number.

In a typical agreement attraction experiment, accuracy decreases when the subject and the distractor mismatches in number (e.g., Bock and Miller, 1991; Eberhard et al., 2005; Staub, 2009). When case is present on the nouns, a matching case (here, +nominative) decreases errors more than a mismatching case (e.g., Nicol and Anton-Mendez, 2009; Hartsuiker et al., 2001, 2003; Badecker and Kuminiak, 2007; Gillespie, 2011; Slioussar, 2018; Gillespie and Pearlmutter, 2013). In Avetisyan et al. (2020), there were four conditions: (a) case marked distractor with a matching number, (b) case marked distractor with a mismatching number, (c) a non case marked distractor with a matching number, and (d) a non case marked distractor with a mismatching number.

Previous results showed a decrease in accuracy in condition (d) vs conditions (a)-(c). In all the models implemented in this work, this decrease in accuracy was assumed before choosing a prior. (a)-(c) conditions were assumed to have similar accuracy.

Its important to emphasise that the priors were chosen such that the predictions of the model resembled each other. This criteria was adapted from previous model comparisons in sentence processing (e.g., Yadav et al., 2022; Husain et al., 2025). We assumed a roughly 10% decrease (this decrease reflects the case effect) in accuracy in condition (d) in comparison to conditions (a)-(c) across models while having enough room to infer from the data. Although note that this number is not chosen through a meta analysis but considering accuracies previously reported in agreement attraction studies (See Table 7 in Kandel et al., 2022).

2.1 Similarity-based interference during Retrieval: Cue-based Retrieval theory (Lewis and Vasishth, 2005)

The first account we consider is the cue-based retrieval account (Lewis and Vasishth, 2005) implemented under the ACT-R architecture (Anderson, 1996; Anderson et al., 2004). As discussed in the previous section, a cue-based retrieval account can explain the effect of case on agreement attraction. For our modeling we implementation the Lewis and Vasishth (2005) and adapted it around the two conditions shown in Figure 2a&b. Specifically, condition (a) and (b) in Table 2 resembled Figure 2a, condition (d) in Table 2 resembled Figure 2b. Condition (c) from Table 2 was taken to have similar probabilities of accuracy as condition (a) and (b) in Table 2. This is because the distractor had the same features as the the subject so the accu-

Model	Locus of interference	Type of interference
Cue-based retrieval	Retrieval	Similarity-based interference
Feature migration theory	Encoding	Similarity-based interference
Case as markers for prediction	Encoding	Prediction distortion

Table 1: Models of subject-verb agreement errors and their locus of interference.

Table 2: Experimental conditions in Avetisyan et al. (2020)

	Case marked	**Non case marked**
Matching number	(a)	(c)
Mismatching number	(b)	(d)

racy reflected by the experiment can be assumed to be the addition of the probability of correct agreement from the subject and the probability of any incorrect agreement from the distractor.

As discussed in the previous section, under a cue-based retrieval theory, chunks receive an activation boost from retrieval cues. The most activated chunk is then retrieved for agreement computation.

The activation boost S_i for chunk i is given by Eq. 1, where W_j is the weight of cue j. We assumed two cues, each with a weight of 0.5. S_{ji} denotes the strength of association between cue j and chunk i, given by Eq. 2, where MAS is the maximum associative strength set to 1 and fan_j is the number of chunks that cue j matches with.

$$S_i = \sum_j W_j S_{ji} \tag{1}$$

$$S_{ji} = MAS - \ln(fan_j) \tag{2}$$

fan_j is where the difference between the +nominative (Figure 2b) and -nominative (Figure 2a) conditions is implemented. In Figure 2a, where the distractor is -nominative, the fan_j for the +nominative cue is 1, since only the subject matches this cue. In Figure 2b, where the distractor is +nominative, the fan_j is 2, since both the subject and the distractor match this cue. This brings down the activation of the subject chunk in the +nominative distractor condition relative to the -nominative condition, increasing the likelihood of misretrieval.

Finally, the probability of retrieving the subject chunk i is given by Eq. 3, where S_i[1] is the activation of the target chunk and s is the noise parameter. The threshold parameter τ had a prior distribution of $\mathcal{N}(0.214, 0.1)$. Recall that this prior is chosen to have condition (d) show $\sim 10\%$ decrease from condition (a)-(c). The predictions of the cue-based model with the given prior is shown in Figure 3.

$$P_{\text{Accuracy}} = \frac{1}{1 + e^{-\frac{(S_i - \tau)}{s}}} \tag{3}$$

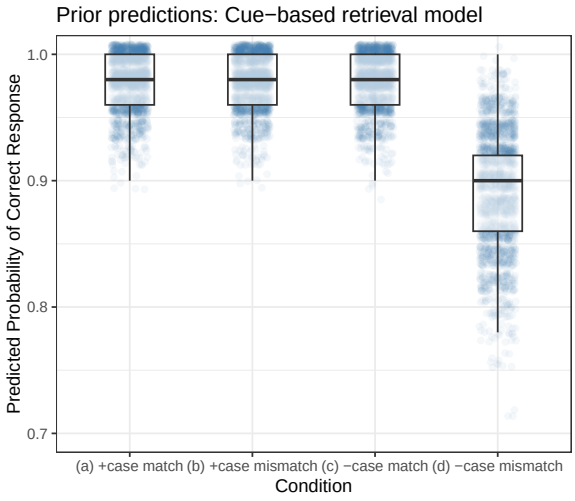


Figure 3: Possible accuracies for condition (a)-(d) in Table 2 under a cue-based retrieval theory.

The retrieval time RT_i for chunk i is given by Eq. 4. Here, F is a scaling parameter with a prior $\mathcal{N}_{lb=0.005}(0.15, 0.05)$ similar to previous implementations of cue-based retrieval model during sentence processing (e.g., Engelmann et al., 2019; Yadav et al., 2022).

$$RT_i = Fe^{-S_i} \tag{4}$$

Assuming a base reaction time β_0 needed for other processes such as integration and motor movement, the overall reaction time is given by Eq. 5.

$$RT_{total} = \beta_0 + RT_i \tag{5}$$

[1]Generally the activation of chunk is present at this part of the equation. As we set other contributing factors such as base level activation to zero, we just use S_i here.

2.2 Similarity-based interference before Retrieval: Feature migration theory (Oberauer and Lang, 2008)

Another model proposed by Avetisyan et al. (2020) to explain the case effects is encoding interference (Nairne, 1990, 2006; Oberauer and Kliegl, 2006). Such models have previously been shown to account for effects seen during sentence processing (e.g., Husain et al., 2025). Similar to the cue-based retrieval model, this model assumes similarity-based interference although at the encoding stage. In terms of case markings, when the subject and the distractor has similar case, the interference increases, thereby decreasing accuracy. In contrast to the cue-based retrieval model, this interference is active during the encoding phase. We diverge from (Avetisyan et al., 2020) by assuming a feature migration model (Oberauer and Lange, 2008) rather than a feature overwriting model. The feature migration theory gives us the possibility that accuracy can decrease in condition (d) relative to other condition. On the other hand, the feature overwriting theory posits similar accuracies (See appendix B).

We use Husain et al. (2025)'s implementation of the feature overwriting theory (Oberauer and Kliegl, 2006) as both feature migration as well as feature overwriting can be defined under a similar mathematical implementation (Oberauer and Lange, 2008). Under this implementation, it is assumed that items are stored in memory and are a source of interference to each other. This interference is modulated by the number of similar features between the items.

In the feature migration theory, interference in and of itself is defined as similarity-based feature migration, wherein, when items in memory are similar, a dissimilar feature can migrate from one item to the other. Each item is assigned an amount of Activation which goes down with increasing number of similar features with other nouns. The activation of the subject noun is given by Eq. 6. Here A_i is the activation of the subject i, n is the number of features of the subject i which is assumed[2] to be 2 in our implementation. r is the rate of feature migration due to similarity-based interference with a prior set to $\mathcal{N}_{lb=1}(1.8, 0.3)$.

[2]We take this assumption largely for simplicity as well as for consistency with the cue-based retrieval model which also assumes 2 features participating in the cue-based retrieval process mentioned in §2.1.

$$A_i = \left(1 - r\frac{C}{2n}\right)^{n-1} \tag{6}$$

The probability of a veridical agreement is a logistic function of the activation of the subject noun A_i defined as Eq. 7. Here, a_0 is the base level activation of the subject set to 1 and T is a noise parameter with a prior set to $\mathcal{N}_{lb=0.01}(0.25, 0.03)$. An increase in noise parameter T shifts the probability of accuracy to near chance level in this model.

$$P_{\text{Accuracy}} = \frac{1}{1 + e^{-\left(\frac{A_i - a_0}{T}\right)}} \tag{7}$$

Recall that the priors were chosen so that the model predicts a similar pattern of probabilities as the cue-based retrieval model across the 4 conditions (See Figure 4).

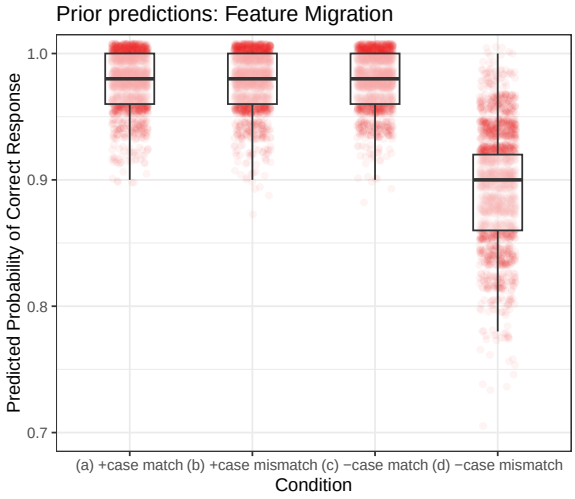


Figure 4: Possible accuracies for condition (a)-(d) in Table 2 under a Feature migration theory.

2.3 Case as Markers for Agreement Prediction

Language comprehension can potentially involve use of case markers to predict syntactic frames (e.g., Husain et al., 2014; Henry et al., 2017; Chow et al., 2018; Zafar, 2025; Yamashita, 1997). Avetisyan et al. (2020) propose that in Armenian, prediction of agreement features is contingent on case markers. The forced choice task employed in Avetisyan et al. (2020) has both a comprehension and a production element. We specifically assume that, when the subject is encountered, the system predicts the agreement features of the matrix verb based on the subject's features. Whenever an appropriate

agreement site appears, these predicted features are utilised for agreement computation.

This process can clearly be present in comprehension. During production, we assume that the system can predict subject-verb agreement as soon as the subject is produced. Similarly, the system can also predict distractor-verb agreement (whenever possible) as soon as the distractor is produced. Further, we also assume that case markers denote whether such a prediction was made.

Overall, these predictions are maintained in memory and are therefore susceptible to interference from other predictions held in memory (cf. Yadav et al., 2020; Husain et al., 2025). Specifically, when a distractor's predicted agreement (when the distractor also has a nominative case) carries different number feature from the subject's, the subject's predicted agreement features can be distorted. When the distractor shares the same features as the subject, no such distortion occurs.

We implement this as follows. The degree of interference S_i from a distractor's prediction is given by Eq. 8, where p_i represents whether the distractor generates a prediction at its clause (1 if yes, 0 if no) and r is the rate of distortion, which is non-zero only when the distractor carries a different agreement feature than the subject. We set r to lie within $\mathcal{N}_{lb=0}(1.73, 0.3)$.

$$S_i = p_i \times r \tag{8}$$

The probability of retaining the veridical representation of the subject's agreement features is given by Eq. 9, where u determines the accuracy of maintaining subject prediction without any interference from distractor prediction.

$$P_{\text{Accuracy}} = \frac{1}{1 + e^{u - S_i}} \tag{9}$$

With our priors, the model roughly predicts similar probabilities as the cue-based retrieval model and the feature migration theory (See Figure 5).

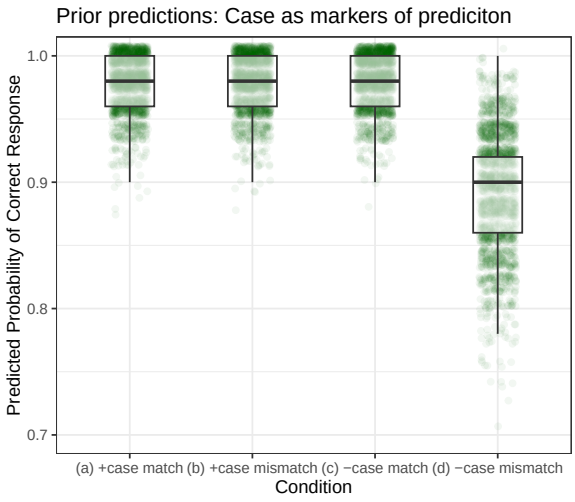


Figure 5: Possible accuracies for condition (a)-(d) in Table 2 under a case as markers for agreement prediction model.

2.4 A theory of agreement for non cue-based retrieval models

Recall that in a cue-based retrieval model the mechanism of retrieval is central to the theory of how agreement operates, i.e., agreement process involves retrieval of subject features. This process of retrieval then manifests as reaction time. On the other hand, there is no process posited at the agreement site for non cue-based retrieval models (feature migration model in 2.2 and case as markers for agreement prediction model in 2.3). This is because these models have not been previously associated with such theories in production.

We assume that items in memory get distorted in both non-cue based retrieval models discussed in section 2.2 and 2.3. This naturalistically demands the processor to infer the correct agreement at the agreement site given the representation of either the item as mentioned in section 2.2 or prediction mentioned in 2.3 is not intact. Such an assumption is consistent with the noisy channel inference model implemented for sentence production (Ryskin et al., 2021).

Furthermore, we assume that probabilities posited in Eq. 7 or Eq. 9 determine the effort required to infer the veridical agreement from a potentially distorted item/prediction, i.e., when an item/prediction is less probable, the effort required is higher, leading to higher reaction times. Such relations between probability and effort has previously been posited during sentence processing (e.g., Levy, 2008, 2013). We implement an information theoretic idea of inference effort I_i in bits; see Eq. 10. Here P_{Accuracy} is probability that the correct agreement prediction has been maintained in memory.

$$I_i = -\frac{\log(P_{\text{Accuracy}})}{\log(2.0)} \tag{10}$$

Finally, the reaction time RT_i to produce agreement from prediction i is given by Eq. 11, where

β_0 represents the base reaction time for processes such as integration and motor execution, and β_1 is a scaling parameter that maps recovery effort in bits to reaction time. The β_1 was set to fall within $\mathcal{N}_{lb=0}(0, 1)$.

$$RT_i = \beta_0 + (\beta_1 \times I_i) \tag{11}$$

Along with the priors for the probabilities, it would also be ideal to have similar reaction time predictions between all three models. We, unfortunately, couldn't achieve that. Hence, our priors were mostly designed around the probabilities. A sensitivity analysis for the cue-based retrieval model with β_0 taken as 0.15, 0.2 and 0.3 found no meaningful difference in their ΔELPD values. Further more, keeping the β_0 value as a free parameter gave us unreasonable estimates (more than 0.35 or 350 ms), often absorbing the effect of other parameters. Hence, we fixed β_0 to 0.2 (0.2 seconds or 200 milliseconds) across models for model comparison.

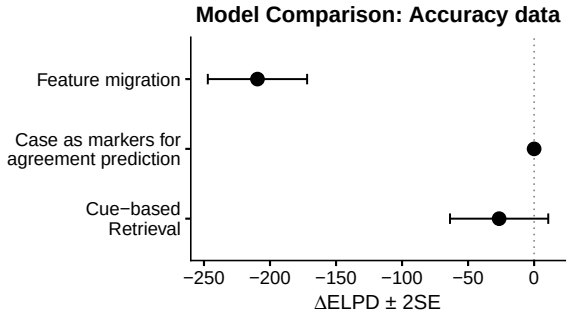


Figure 6: Expected Log Pointwise Predictive Density (ΔELPD) differences across models for the accuracy data.

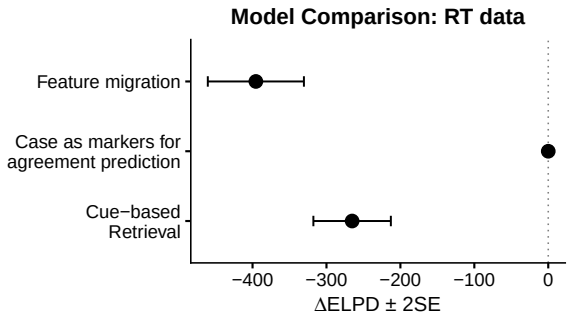


Figure 7: Expected Log Pointwise Predictive Density (ΔELPD) differences across models for the reaction time data.

3 Model Comparison

All models were implemented in Stan (Stan Development Team, 2026) and fitted using Bayesian inference via the `rstan` package in R[3]. The data comprised two dependent measures: accuracy (whether the verb was produced with veridical agreement) and reaction time. Each observation was coded for two experimental factors, case (match vs. mismatch) and number of the attractor (singular vs. plural), yielding four conditions.

Each of the three models were fitted to both accuracy and reaction time data simultaneously. All models were fitted using four parallel chains.

To compare models, we used leave-one-out cross-validation method as implemented in the `loo` package in R. LOO estimates the expected log pointwise predictive density (ELPD), which quantifies how well a model predicts held-out data. A higher ELPD indicates better predictive performance. Pareto k diagnostic values were inspected for each model to ensure the reliability of the LOO estimates (Nicenboim et al., 2025).

Model comparison was carried out separately for accuracy and reaction time. For each dependent measure, the three models were compared using the `loo_compare` function, which ranks models by their ELPD and provides the difference in ELPD along with its standard error. A model with more than 2 SE difference in ELPD is taken to have better out-of-sample predictive performance relative to the others.

For accuracy, the case as markers for agreement prediction model with inference performed best. The cue-based retrieval model was the next best with a ΔELPD of -26.5 (SE = 18.6). This difference is within two standard errors, suggesting that the two models are not reliably distinguishable in their predictive performance for accuracy. The feature migration model, however, performed substantially worse with ΔELPD values of -209.5 (SE = 18.8). This difference is well beyond two standard errors, indicating reliably poorer predictive performance.

For reaction time, the case as markers for agreement prediction model with inference was again the best model. Unlike accuracy, the difference between the case as markers for agreement prediction model and the cue-based retrieval model (ΔELPD = -265.3, SE = 26.2) was well beyond two standard errors, indicating that the case as markers

[3]Codes are available at this link.

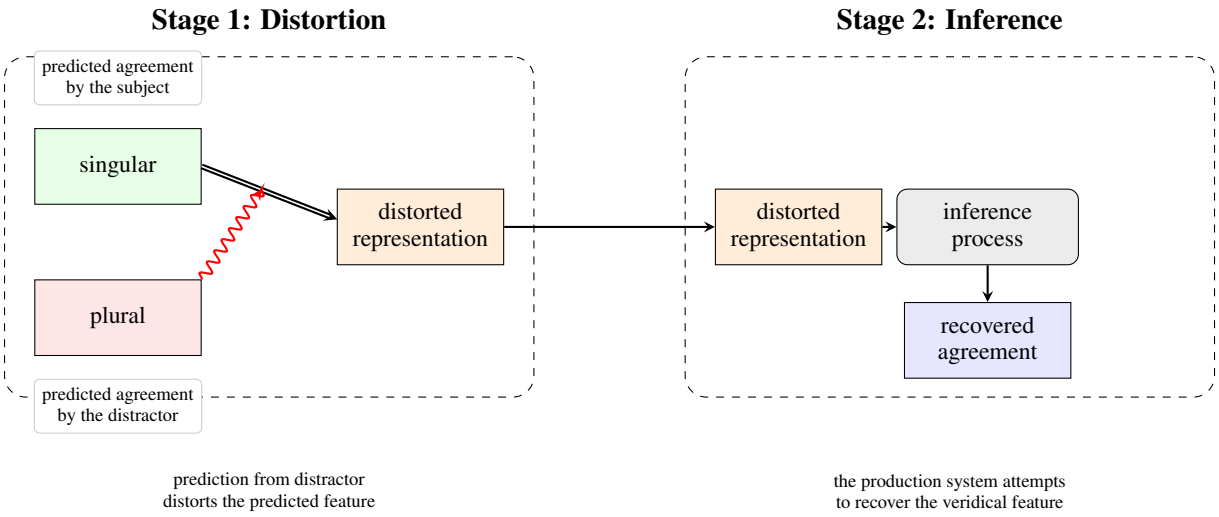


Figure 8: A schematic of the two-stage model. In Stage 1, the predicted agreement feature is susceptible to distortion from interference by the predicted agreement for the distractor clause. In Stage 2, the production system infers the veridical agreement feature from the potentially distorted representation. The effort required for this inference determines the reaction time

for agreement prediction model reliably outperforms the cue-based retrieval model for reaction time. The feature migration model with inference still performed the worst (ΔELPD = -395.5, SE = 32.5).

4 General Discussion

The current paper achieves two goals. First, it implements three distinct computational models that can explain the role of case during agreement production, focusing on the phenomenon of agreement attraction.

Second, it compared these different models and found that a 'case as markers for agreement prediction' model followed by an inference component explains the effect of case-marker during agreement in Armenian better than a cue-based retrieval model. Specifically, our model comparison results illustrate a language production system where agreement computation is largely predictive in nature. These predictions are in turn susceptible to distortion in memory; given such distortions, the processor infers the original agreement features (See Figure 8). Additionally, our results highlight that using both accuracy as well as reaction time data provides greater insights to understanding agreement processes (also see, Staub, 2010).

4.1 Prediction and incrementality

The scope of planning during sentence production is thought to be minimal in nature (Kempen and Hoenkamp, 1987; Levelt et al., 1999; Levelt, 1999). While some proposals argue that the scope of planning is radically incremental (e.g., Konopka, 2012; Paul, 1970; Bock and Irwin, 1980; Ferreira, 1994), others have argued that the scope of planning is larger (e.g., an entire clause, a verbal unit, etc.)(e.g., Momma et al., 2020; Momma and Yoshida, 2023; Bock and Levelt, 1994; Bock and Cutting, 1992). There is very little work that has explored agreement computation in the context of planning scope. The current work contributes towards this discussion as well.

Our results provide some evidence for a production system where the agreement representation is in fact planned (via prediction). Specifically, our results point to a scenario where the decision to predict agreement is contingent on the use of nominal case-marker. In other words, the presence of a nominative case marker on a noun triggers a prediction that agreement will need to be computed within that clause. When a distinct prediction is generated by the distractor, it interferes with the agreement prediction generated by the subject, giving rise to attraction errors. The absence of a nominative case marker, on the other hand, does not generate such a prediction, thereby making the inference process easier. As such, while the current work suggests that agreement representations are planned prior to the articulation of the verb, the precise nature of such representations needs to be investigated further.

4.2 Cross-linguistic validity

An important point needs to be made regarding the cross-linguistic validity of our case as markers for agreement prediction model. The case as markers for agreement prediction model entails that whenever there is a nominative case, a prediction for agreement is made. This holds for a language like Armenian but need not be true for other languages where a nominative noun can appear without any agreement within its clause. For example, in the Slovak (shown in example 1), having a nominative case would not automatically mean that there will be agreement within its clause. Hence, the case as markers for agreement prediction model only stays relevant for languages where nominative case predicts agreement. In other words, how language properties of a language dictate the prediction mechanism need to be investigated in future studies.

Regardless, the case as markers for agreement prediction model can also apply to other features which predict agreement. For example, in Hindi, a distractor-induced agreement attraction has been attributed to a 'feature' of the distractor when it has agreement within its clause (See controller coding hypothesis in Bhatia and Dillon, 2022). Interestingly this presence of a feature that cause interference and potential for agreement prediction is confounded in their explanation. In other words, a predictive system could also explain this data. Within a prediction based account, whenever a prediction for agreement is made at the distractor, inference of the subject agreement becomes difficult. This can effectively replace the 'feature' which was previously attributed in Bhatia and Dillon (2022) to cause interference. Overall, these cross-linguistic considerations suggest that while the specific cue that drives prediction may vary across languages, the underlying mechanism of prediction followed by inference may be a more general property of the agreement production system in languages with similar case and agreement properties.

4.3 Agreement attraction during production

The agreement attraction phenomenon during production has generally been evaluated either under an encoding based interference account (e.g., Bock and Miller, 1991; Bock et al., 2001; Eberhard et al., 2005; Franck et al., 2002) or a retrieval based account (e.g., Badecker and Kuminiak, 2007; Bhatia and Dillon, 2022; Kandel et al., 2022). Our results suggests that apart from a simple interference/encoding-based constraints, the role of prediction and inference for such computations are equally important (cf. Ryskin et al., 2021), if not more. On the other hand, there is a strong evidence for the interference/encoding-based constraints on agreement computation during sentence comprehension (for an overview see Yadav et al., 2022). This suggests that agreement computation processes during comprehension might be distinct from how such computations happen during sentence production.

5 Conclusion

The current paper investigated the role of case during agreement attraction in sentence production. Through computational modeling, we compared different accounts that could explain the effect of case on agreement errors. Our results show that a case as markers for agreement prediction followed by an inference component provides a better explanation than a cue-based retrieval model, which was previously argued to account for such effects.

Overall, this work puts forward a two-stage account of agreement production, comprising prediction and inference, that advances our understanding of how case information modulates agreement errors. Future work comparing the current models as well as additional models across languages with different case-agreement configurations will be critical in evaluating the generality of this account.

6 Limitations

6.1 Case vs. Subjecthood

One important caveat regarding Avetisyan et al. (2020)'s production data is that every time the distractor case was nominative, it was also the subject within its own clause. On the other hand, when it was not nominative, it was an object within its own clause. Critically, objects never show agreement in Armenian. Such a confound is important to address because subjecthood can impact agreement attraction in some languages, e.g., Romanian, (Bleotu and Dillon, 2024), but not in others, e.g., Hindi, (Bhatia and Dillon, 2022). Regardless of whether this effect was driven by case or subjecthood, the case as markers for agreement prediction model can be adapted for both scenarios. Subjecthood might mark a prediction of agreement in place of case in Armenian. As mentioned in §4.3, the current prediction based theory can also extend to

Hindi (Bhatia and Dillon, 2022). In Hindi, instead of case or subjecthood, whether the distractor had agreement within its own clause predicted agreement attraction. It would be insightful to compare these models on Hindi as well as Romanian data in future work.

6.2 Role of Paradigm

A forced choice paradigm as implemented in Avetisyan et al. (2020) has both a comprehension as well as a production component. Specifically, the rapid serial visualization of words constitutes the comprehension aspect of the task while the choice for agreement constitutes the production aspect. An important test for these results would be to replicate these results through other paradigms (e.g., preamble repetition paradigm, scene description paradigm).

6.3 A note on condition (c)

While agreement errors during comprehension have previously been implemented computationally, models of agreement errors during production have not. The present work addresses this gap and, as such, stands as a starting point for future computational modeling of agreement errors in production. This framing has necessarily entailed certain simplifications, most notably in the implementation of condition (c). As discussed in §2, we assumed the same accuracy for condition (c) as for conditions (a) and (b), and further assumed that the underlying process was similar across the three conditions. This choice was motivated by the desire to maintain a consistent set of assumptions applicable across all models, particularly in the absence of clear empirical or theoretical guidance during implementation. In future work, we plan to develop models that more faithfully/fruitfully capture the processes specific to condition (c).

References

John R. Anderson. 1996. ACT: A simple theory of complex cognition. *American Psychologist*, 51(4):355–365. Place: US.

John R. Anderson, Daniel Bothell, Michael D. Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. 2004. An Integrated Theory of the Mind. *Psychological Review*, 111(4):1036–1060. Place: US.

Serine Avetisyan, Sol Lago, and Shravan Vasishth. 2020. Does case marking affect agreement attraction in comprehension? *Journal of Memory and Language*, 112:104087.

William Badecker and Frantisek Kuminiak. 2007. Morphology, agreement and working memory retrieval in sentence production: Evidence from gender and case in Slovak. *Journal of Memory and Language*, 56:65–85.

Sakshi Bhatia and Brian Dillon. 2022. Processing agreement in Hindi: When agreement feeds attraction. *Journal of Memory and Language*, 125:104322.

Adina Camelia Bleotu and Brian Dillon. 2024. Romanian (subject-like) DPs attract more than bare nouns: Evidence from speeded continuations. *Journal of Memory and Language*, 134:104445.

J. Kathryn Bock and David E. Irwin. 1980. Linearization in sentence production: Evidence from comprehension. *Journal of Verbal Learning and Verbal Behavior*, 19(4):467–479.

Kathryn Bock and J. Cooper Cutting. 1992. Regulating mental energy: Performance units in language production. *Journal of Memory and Language*, 31(1):99–127. Place: Netherlands.

Kathryn Bock and Kathleen M. Eberhard. 1993. Meaning, sound and syntax in english number agreement. *Language and Cognitive Processes*, 8(1):57–99.

Kathryn Bock, Kathleen M. Eberhard, J. Cooper Cutting, Antje S. Meyer, and Herbert Schriefers. 2001. Some Attractions of Verb Agreement. *Cognitive Psychology*, 43(2):83–128.

Kathryn Bock and Willem JM Levelt. 1994. Language production: Grammatical encoding. In Morton Ann Gernsbacher, editor, *Handbook of psycholinguistics*, pages 945–984. Academic Press.

Kathryn Bock and Carol A Miller. 1991. Broken agreement. *Cognitive Psychology*, 23(1):45–93.

Wing-Yee Chow, Andrew Nevins, and Manuel Carreiras. 2018. Effects of subject-case marking on agreement processing: ERP evidence from Basque. *Cortex*, 99:319–329.

Kathleen Eberhard, John Cutting, and Kathryn Bock. 2005. Making Syntax of Sense: Number Agreement in Sentence Production. *Psychological review*, 112:531–59.

Kathleen M. Eberhard. 1997. The Marked Effect of Number on SubjectVerb Agreement. *Journal of Memory and Language*, 36(2):147–164.

Felix Engelmann, Lena A. Jger, and Shravan Vasishth. 2019. The Effect of Prominence and Cue Association on Retrieval Processes: A Computational Account. *Cognitive Science*, 43(12):e12800.

Victor S. Ferreira. 1994. The choice of structural transitions in oral delivery of pennsylvania dutch. *Journal of Memory and Language*, 33(6):715–736.

Julie Franck, Gabriella Vigliocco, and Janet Nicol. 2002. Subject-verb agreement errors in French and English: The role of syntactic hierarchy. *Language and Cognitive Processes*, 17(4):371–404. _eprint: https://doi.org/10.1080/01690960143000254.

Maureen Gillespie. 2011. Effects of Semantic Integration and Advance Planning on Grammatical Encoding in Sentence Production. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 33.

Maureen Gillespie and Neal J. Pearlmutter. 2013. Against structural constraints in subjectverb agreement production. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 39(2):515–528.

Robert J. Hartsuiker, Inés Antón-Méndez, and Marije Van Zee. 2001. Object Attraction in Subject-Verb Agreement Construction. *Journal of Memory and Language*, 45(4):546–572.

Robert J. Hartsuiker, Herbert J. Schriefers, Kathryn Bock, and Gerdien M. Kikstra. 2003. Morphophonological influences on the construction of subject-verb agreement. *Memory & Cognition*, 31(8):1316–1326.

Nick Henry, Holger Hopp, and Carrie N. Jackson. 2017. Cue additivity and adaptivity in predictive processing. *Language, Cognition and Neuroscience*, 32(10):1229–1249.

Samar Husain, Apurva, Ishita Arun, and Himanshu yadav. 2025. The effect of similarity-based interference on bottom-up and top-down processing in verb-final languages: Evidence from Hindi. *Journal of Memory and Language*, 143:104627.

Samar Husain, Shravan Vasishth, and Narayanan Srinivasan. 2014. Strong Expectations Cancel Locality Effects: Evidence from Hindi. *PLOS ONE*, 9(7):e100986.

Margaret Kandel, Cassidy Rae Wyatt, and Colin Phillips. 2022. Agreement attraction error and timing profiles in continuous speech. *Glossa Psycholinguistics*, 1(1).

Gerard Kempen and Edward Hoenkamp. 1987. An incremental procedural grammar for sentence formulation. *Cognitive science*, 11(2):201–258.

Agnieszka E Konopka. 2012. Planning ahead: How recent experience with structures and words changes the scope of linguistic planning. *Journal of Memory and Language*, 66(1):143–162.

W. J. Levelt, A. Roelofs, and A. S. Meyer. 1999. A theory of lexical access in speech production. *The Behavioral and Brain Sciences*, 22(1):1–38; discussion 38–75.

Willem JM Levelt. 1999. Producing spoken language: A blueprint of the speaker. *The neurocognition of language*, 83:122.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Roger Levy. 2013. Memory and surprisal in human sentence comprehension. In *Sentence processing*, pages 78–114. Psychology Press.

Richard L. Lewis and Shravan Vasishth. 2005. An Activation-Based Model of Sentence Processing as Skilled Memory Retrieval. *Cognitive Science*, 29(3):375–419.

Michael McCloskey. 1991. Networks and theories: The place of connectionism in cognitive science. *Psychological Science*, 2(6):387–395.

Masaaki Momma, L Robert Slevc, and Colin Phillips. 2020. The timing of verb selection in english active and passive sentences. *Journal of Memory and Language*, 111:104066.

Masaaki Momma and Masaya Yoshida. 2023. The architecture of sentence production. *Annual Review of Linguistics*, 9:219–238.

James S. Nairne. 1990. A feature model of immediate memory. *Memory & Cognition*, 18(3):251–269.

James S. Nairne. 2006. Modeling distinctiveness: Implications for general memory theory. In *Distinctiveness and memory*, pages 27–46. Oxford University Press, New York, NY, US.

Bruno Nicenboim, Daniel J Schad, and Shravan Vasishth. 2025. *Introduction to Bayesian data analysis for cognitive science*. CRC Press.

Janet L. Nicol and Ines Anton-Mendez. 2009. The effect of case marking on subject-verb agreement errors in English.

Klaus Oberauer and Reinhold Kliegl. 2006. A formal model of capacity limits in working memory. *Journal of Memory and Language*, 55(4):601–626.

Klaus Oberauer and Elke B. Lange. 2008. Interference in verbal working memory: Distinguishing similarity-based confusion, feature overwriting, and feature migration. *Journal of Memory and Language*, 58(3):730–745.

Hermann Paul. 1970. *Prinzipien der Sprachgeschichte (Principles of Language History)*. Max Niemeyer Verlag. Original work published 1880.

Rachel Ryskin, Leon Bergen, and Edward Gibson. 2021. Agreement errors are predicted by rational inference in sentence processing. Technical report, PsyArXiv. Type: article.

Natalia Slioussar. 2018. Forms and features: The role of syncretism in number agreement attraction. *Journal of Memory and Language*, 101:51–63.

Stan Development Team. 2026. Stan Users Guide.

Adrian Staub. 2009. On the interpretation of the number attraction effect: Response time evidence. *Journal of Memory and Language*, 60(2):308–327.

Adrian Staub. 2010. Response time distributional evidence for distinct varieties of number attraction. *Cognition*, 114(3):447–454.

Shravan Vasishth and Felix Engelmann. 2021. *Sentence Comprehension as a Cognitive Process: A Computational Approach*. Cambridge University Press. Google-Books-ID: 6KZKzgEACAAJ.

Aki Vehtari, Andrew Gelman, and Jonah Gabry. 2017. Practical bayesian model evaluation using leave-one-out cross-validation and waic. *Statistics and computing*, 27(5):1413–1432.

Aki Vehtari, Daniel Simpson, Andrew Gelman, Yuling Yao, and Jonah Gabry. 2024. Pareto smoothed importance sampling. *Journal of Machine Learning Research*, 25(72):1–58.

Gabriella Vigliocco and Tiziana Zilli. 1999. Syntactic Accuracy in Sentence Production: The Case of Gender Disagreement in Italian Language-Impaired and Unimpaired Speakers. *Journal of Psycholinguistic Research*, 28(6):623–648.

Himanshu Yadav, Garrett Smith, Sebastian Reich, and Shravan Vasishth. 2022. Number feature distortion modulates cue-based retrieval in reading. Technical report, PsyArXiv. Type: article.

Himanshu Yadav, Ashwini Vaidya, Vishakha Shukla, and Samar Husain. 2020. Word order typology interacts with linguistic complexity: A cross-linguistic corpus study. *Cognitive science*, 44(4):e12822.

Hiroko Yamashita. 1997. The Effects of Word-Order and Case Marking Information on the Processing of Japanese. *Journal of Psycholinguistic Research*, 26(2):163–188.

Mudafia Zafar. 2025. *Planning scope in production how grammar shapes incrementality*. Ph.D. thesis, Indian Institute of Technology Delhi.

A Codes for the paper

Codes are available at this link.

B Feature migration over feature overwriting

A feature overwriting account assumes that features shared between items in memory get degraded. Such an assumption does not align well with the base assumptions for our conditions. Under a feature overwriting model, the feature being degraded would be case (the shared feature), and not number. This in turn entails no predicted difference in number agreement between condition (c) and condition (d).

A feature migration model, on the other hand, fits better with our assumptions for the conditions. Although both condition (c) and condition (d) involve similarity-based interference, only condition (d) has a dissimilar number feature on the distractor that can migrate to the subject noun. Hence, feature migration model can explain our data but not a feature overwriting model.

C Model flexibility and degrees of freedom

A model is not a theory (McCloskey, 1991). A good fit to the data can come from the flexibility a model's free parameters afford rather than the validity of the underlying mechanism it posits. The three models we compared slightly differ in the number of free parameters they contain. The cue-based retrieval model has two (τ, F), the feature migration model has three (r, T, β_1), and the case as markers for agreement prediction has three (r, u, β_1). Notably, the case as markers for agreement prediction model is not the most flexible of the three in terms of number of parameters. This makes it unlikely that its better performance is solely an artifact of having more degrees of freedom.

Propositional compositionality in neural language models

Jane Li, Abhinav Patil, Kyle Rawlins
Department of Cognitive Science
Johns Hopkins University
{sli213, aptil10, kgr}@jhu.edu

1 Introduction

One of the most fundamental representations in linguistic semantics is that of the proposition (McGrath and Frank, 2005), standardly taken as the carrier of truth-conditions. Recent work shows that some form of truth can be decoded from language models (Azaria and Mitchell, 2023; Li et al., 2023), and strikingly, that for some models, truth is even represented linearly in intermediate layers (Marks and Tegmark, 2024, GoT). We take this line of work a step further and argue that neural language models can use propositional representations compositionally (Janssen 2010; Pickel and Szabó 2025 a.o.), drawing from evidence of the behaviour of logical connectives: the *linear compositionality* hypothesis. Specifically, we show (a) that the truth values of individual conjuncts can be decoded independently of the truth value of a complex conjunction, and (b) that we can causally intervene on individual conjuncts in a way that affects the truth value of the whole.

2 Model and datasets

This work analyzes the sentence representations generated by Llama-3.1-8B (Grattafiori et al., 2024). The methods documented in this work generalize to any neural language model, but we take this as a starting point because previous work (GoT) uses models from the Llama series.

We constructed two datasets (4,000 sentences each) consisting of conjoined sentences in two knowledge domains: `cities` and `kinds`. Each dataset is balanced for the truth value of its atomic parts, as illustrated by Table 1. The atomic sentences for the `cities` dataset were recruited from GoT, whereas `kinds` was manually constructed.

3 Decoding logical expressions

Can we detect a linear encoding of truth – of a sentence's parts and globally – in the model's hidden representation of a sentence? The main finding of GoT suggests that this is possible for simple expressions using mass-mean probing, and here we demonstrate that it is also possible to linearly decode for the truth of the conjuncts and the global truth of a complex sentence.

Dataset	Example sentence
`cities`	The city of {Baltimore, Manila} is in the US, and the city of {Vancouver, Paris} is in Canada.
`kinds`	A {guppy, violin} is a fish, and a {stadium, rose} is a building.

Table 1: Example sentences from each of the datasets. Each item consists of four sentences, spanning all possible truth value configurations.

Mass-mean probing is a linear classification method with the probability function $p(\vec{x}) = \sigma(\theta \cdot \vec{x})$. Crucially θ is a difference vector, defined as the difference between the average positive and negative representation (in our case, model representations of true and false sentences). If the probe is found to be efficacious for classification, then we may hypothesize θ to be well-approximating a sentence's truth representation. This allows us to further test how θ affects a model's computation of truth downstream (§5).

At the final token of the sentence, we decode for the following truth-related properties: truth of the first conjunct, second conjunct, and the global truth value of the sentence. Individual probes are fitted for each of these properties, each layer, and each dataset, resulting in 3 (properties) $\times$ 32 (layers) $\times$ 2 (datasets) = 192 total probes. We also report the decoding of truth of the atomic sentences (conjuncts) that make up the dataset.

Each probe is fitted on 1,000 randomly sampled sentences from the dataset and evaluated on the held-out ones. The reported accuracies are the mean of 100 runs, each sampling a different subset. A probe is reported have above-chance signal if

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 365–368
July 3-4, 2026 ©2026 Association for Computational Linguistics

the accuracy distribution is significantly greater than null accuracy distribution, measured from null probes where the labels for truth are shuffled in the fitting sentence set.

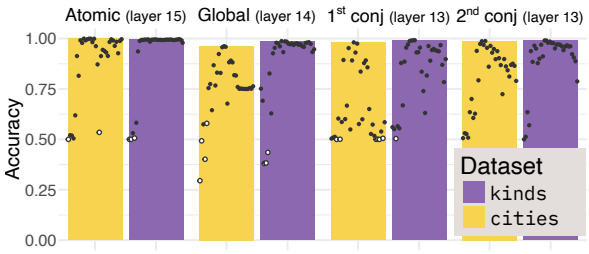


Figure 1: Decoding accuracies (of the layer with maximum signal) on held-out sentences on various truth properties of the sentence. Dots on the bar represent accuracies on each other layer, organized by layer depth (left-to-right). Hollow dots represent decoding accuracies that are at-chance.

We find that beyond early layers, mass-mean probes reliably detect various truth values of both datasets, though with substantial variation in their accuracy across layers and datasets. For each type of truth-related property, we find a layer that classifies held-out sentences at near-perfect accuracies (Fig. 1). PCA plots of the top PCs (Fig. 2) also reflect this separation.

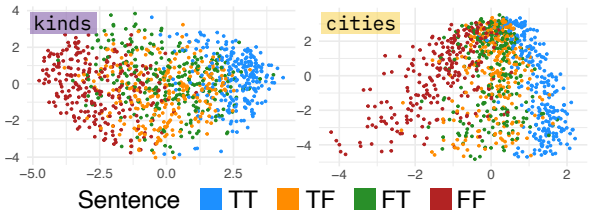


Figure 2: Top two PCs of sentence representations at layer 14, coloured by the truth value combinations of the sentence.

4 Whole state interventions

The previous experiment suggests that the model has some linear encoding of truth, but the linear compositionality hypothesis makes stronger predictions beyond successful decoding: we expect compositional representations of conjunct truth to be relevant for the computation of global truth. In this experiment, we examine a specific manifestation of this prediction: that computations over specific hidden states are responsible for taking conjunct truth as input and determining global truth via their outputs. This experiment demonstrates through hidden state replacement within minimal pairs that these states exist and are relatively localized.

4.1 Prompting for truth

First, we define a query that a model reliably correctly responds to. Here we use a mixture of example and question prompting, embedding each target sentence in the frame: `[ex. true sentence] Is this statement true? Yes. [ex. false sentence] Is this statement true? No. [target sentence] Is this statement true?`. We then measure the probabilities and rank of `Yes` or `No` as a continuation of the prompt. We find that this prompt is 91.0% accurate across all sentences we study, with some variation between datasets and conjunct truth value combinations (Fig. 3). We apply these measurements to truth-altered sentences to observe how the model's credence of the sentence fluctuates as a result of an alteration.

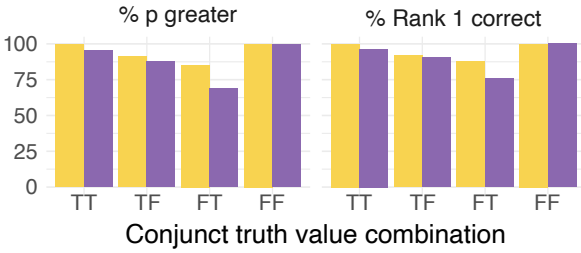


Figure 3: Prompting for `Yes` vs. `No` as a proxy for model-assigned credence to the sentence. (**A**; left) The % of items that correctly assign `Yes`/`No` as its more likely continuation over the alternative. (**B**; right) % of items that assign the correct token as the most likely response.

4.2 Hidden state swapping

We intervene on the truth value of the first conjunct of False-first conjunct sentences (FF and FT) by replacing a hidden state with the matching hidden state of its True-first counterpart. The main measurement is the degree of relative probability increase for `Yes` as a continuation to the prompt.[1] We conducted the swap over all layers and at five increments at the end of the first conjunct: `is`, `in/a`, `country/category`, `,` (comma), and `and`.

The linear compositionality hypothesis predicts that sentences that change in their hypothesized global truth value due to this alteration (i.e., FT → TT; Critical) should elicit greater probability increases relative to their counterparts that do not trigger a global truth value change (e.g., FF → TF; Control). We do not expect any effects of truth value alterations to arise in the first two timepoints

[1]Here we define relative probability (RelP) of `Yes` as $\frac{P(\text{Yes})}{P(\text{Yes})+P(\text{No})}$. The main measurement, relative probability increase, is the difference between RelP of the original sentence and RelP of the altered sentence.

(is, in/a) because the truth of the conjunct is indeterminate at that point of the sentence, thus acting as a methodological control.

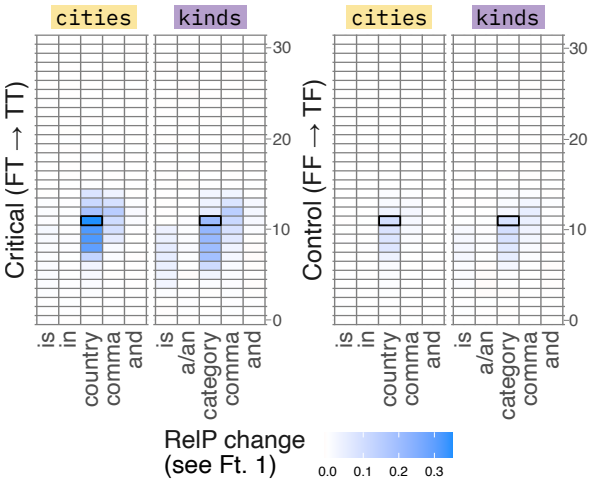


Figure 4: Relative probability change as a function of a state swap, where states vary by layer (x-axis) and token (y-axis). The highlighted grid indicates the hidden state we conduct follow-up analyses on.

We find that sentences in the Critical condition exhibit a greater increase in probability than the Control condition, though the Control condition also elicited positive probability change (Fig. 4). These results are mirrored with change in rank as the dependent variable. The results suggest that a model makes reference to the truth of the first conjunct when computing the truth or falsity of the larger sentence. This reference computation is relatively localizable, concentrated in mid-early layers (8-12) at the country or category tokens (for `cities` and `kinds` respectively). However, probability shifts in the Control condition also indicate that downstream measures such as output probabilities are influenced by conjunct truth, independent of its contribution to global truth. We also find stronger probability shifts for the `cities` dataset than the `kinds` dataset. The `kinds` dataset elicits probability increases in the control timepoints (specifically at the token `is`); we return to this in the discussion.

5 θ-intervention over target states

The findings so far set us up to test the linear compositionality hypothesis directly: if the linear encoding of truth θ (§3) well-approximates an abstract representation of conjunct truth in sentence representation space, then we expect intervening on a causally relevant hidden state by θ should impact model response to queries. At that target hidden state identified in §4, we computed θ for each dataset separately. Instead of replacing the hidden states, we now add or subtract (for False-first and True-first sentences respectively) the value θ from the hidden state. We then conduct the same types of measurements (relative probability change, rank change) as §4 on the altered sentences.

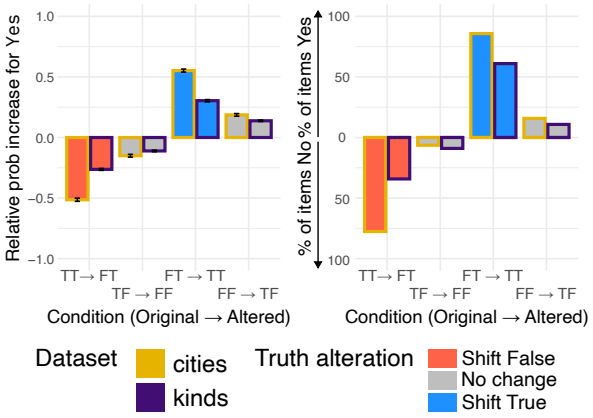


Figure 5: Results from the θ-intervention experiment. All results here are from alterations conducted on layer 11. (**A**; left) The degree of relative probability increase or decrease to the continuation `Yes`. Error bars indicate the 95% CI of the mean. (**B**; right) % of sentences that have changed their ranking to `Yes` (upward) or `No` (downward) as a result of the intervention.

We find that θ-intervention elicits similar results as whole state swapping: greater increases in the Critical condition, while observing some non-negligible increase in the Control condition (Fig. 5A, B). These results are also mirrored in the case of altering True-first sentences, where we observe an increase in the probability of `No` responses post-alteration. Once again, these results support the linear compositionality hypothesis while demonstrating that there is an independent contribution of conjunct truth to downstream tasks. Additionally, for both measures, the effect magnitude on probability (6.02%, $t = 21.035$, $p < 0.01$) are greater relative to whole-state intervention. This further strengthens the argument that the encoding of abstract conjunct truth, as approximated by θ, is the main driver of global truth computations.

6 Discussion

This set of experiments provide preliminary evidence for the linear compositionality hypothesis. We demonstrate that it is possible, for massively distributed systems without an explicit encoding of propositions, truth, or conjunction, to be well-approximated by the compositional, symbolic systems devised by semanticists and logicians. This

hypothesis ideally should extend to other truth-related computations such as other logical connectives, and should be extended to examine the computations that a model is carrying out, not just the input and output representations. In some preliminary work in disjunctive sentences, we find that the hypothesis falls apart. There are many linguistically-interesting hypotheses that may explain the difference between these logical operators, and we plan to explore this in future work.

6.1 Dataset-level variation

Across experiments, we observe differences between datasets: `kinds` shows considerably stronger decoding effects, but this relationship is reversed for the other experiments. Our working hypothesis is that this is due to the two datasets varying in the frequency to which the conjuncts are explicitly mentioned in some form in the training data. <City, country> pairs from `cities` dataset are likely to be stated in the training data frequently, or in the form of structured infoboxes, whereas commonsense facts from `kinds` may be less frequent in the training data even though they represent widely agreed upon knowledge (Gordon and Van Durme, 2013). However, this hypothesis does not explain why these differences cause performance differences between tasks.

6.2 Truth, uncertainty, and plausibility

In this work, we employed a "neat" definition of truth and used examples that are easily categorizable and verifiable under this notion of truth, as our goal was to study the computations of conjunctive sentences. As follow-up work, it would be important to expand to various more graded effects that are known to impact both human and model truth-value judgements, such as uncertainty (Hu et al., 2025a) and sentence plausibility (Hu et al., 2025b).

6.3 Limitations

We highlight two major limitations of this work. Following prior work, we use prompting as a measure of model 'credence'; in doing so we find fluctuations in accuracies across different truth value combinations (Fig. 3), potentially confounding any intervention experiments that also rely on comparisons across conditions. More investigation of the relationship between internal model states and prompting behavior is crucial. A second shortcoming concerns the treatment of token-level interventions, which can sometimes mismatch linguistic units. It remains unclear if failure of intervention at a single token occurs because of a failure to represent truth, or an unexpected positional representation of truth relative to the token sequence.

References

Amos Azaria and Tom Mitchell. 2023. The Internal State of an LLM Knows When It's Lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, AKBC '13, pages 25–30, New York, NY, USA. Association for Computing Machinery.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models. *arXiv preprint*. ArXiv:2407.21783 [cs].

Jennifer Hu, Michael A. Lepori, and Michael Franke. 2025a. Signatures of human-like processing in Transformer forward passes. *arXiv preprint*. ArXiv:2504.14107 [cs.AI].

Jennifer Hu, Felix Sosa, and Tomer Ullman. 2025b. Shades of zero: Distinguishing impossibility from inconceivability. *Journal of Memory and Language*, 143:104640.

Theo M.V. Janssen. 2010. Compositionality. In Johan van Benthem and Alice G. B. ter Meulen, editors, *Handbook of Logic and Linguistics*. Elsevier.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-Time Intervention: Eliciting Truthful Answers from a Language Model. *Advances in Neural Information Processing Systems*, 36:41451–41530.

Samuel Marks and Max Tegmark. 2024. The Geometry of Truth: Emergent Linear Structure in Large Language Model Representations of True/False Datasets. In *First Conference on Language Modeling*.

Matthew McGrath and Devin Frank. 2005. Propositions. In Edward N. Zalta, editor, *Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University.

Bryan Pickel and Zoltán Gendler Szabó. 2025. Compositionality. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*, Winter 2025 edition. Metaphysics Research Lab, Stanford University.

Can language models process linguistic deference?

Youngdong Cho
Cornell University
yc2544@cornell.edu

Chloe Dokyung Kwon
Cornell Univeristy
dk837@cornell.edu

Abstract

Honorifics are linguistic forms that encode respect toward a socially valued individual or entity. This paper investigates how language models process Korean subject honorifics, which signal the social status of the subject through specific morphological markers. We evaluate a set of language models to determine whether they process honorifics in a human-like way by capturing the socio-pragmatic constraints governing their use, rather than merely relying on surface co-occurrence patterns. Our results indicate a systematic dissociation: models generally succeeded in detecting surface morphosyntactic mismatches, successfully treating unacceptable honorific constructions as less expected. However, models consistently favored overt honorific marking regardless of the subject's social status, suggesting reliance on surface heuristics over genuine pragmatic knowledge. These findings suggest that language models have not fully acquired the socio-pragmatic constraints underlying honorific use, even when extensively trained on Korean text.

1 Introduction

Honorifics are linguistic forms that convey deference towards a socially respected referent, typically a person of superior social standing. This paper aims to see how language models process honorifics, focusing on *subject honorifics*, which reflect the relationship between the speaker and the subject of the sentence. Subject honorifics are realized through special morphological inflections attached to the subject and the verb when the subject is an honorable entity, as illustrated in (1b).[1]

(1) a. Ai-ka pyenci-lul ssu-ess-ta.
child-NOM letter-ACC write-PST-DECL
'The child wrote a letter.'

b. Kyoswunim-**kkeyse** pyenci-lul
professor-**NOM.HON** letter-ACC
ssu-**si**-ess-ta.
write-**HON**-PST-DECL
'The professor wrote a letter.'

Korean is an agglutinative language in which grammatical roles, such as subject and object, are indicated by case markers attached to nouns. The subject is marked with a nominative marker *-i/ka*, and the object is marked with an accusative marker *-ul/lul* as shown in (1a). However, when the subject is an honorable entity such as *kyoswunim* 'professor' in (1b), the plain nominative marker *-i/ka* can be replaced by an honorific nominative marker, and the verb may carry the honorific suffix *-si* after the verbal stem.

At first glance, Korean subject honorifics resemble English subject-verb agreement in that the subject triggers specific morphological inflection on the verb (Linzen et al., 2016). However, the Korean system exhibits a more complex pattern; unlike the obligatory nature of English agreement, honorific markings in Korean remain optional, even though the subject is an honorable entity (Noh et al., 2024). For example, the verbal honorific marker *–si* can be omitted, and the nominative marker can alternatively be realized as the plain marker *–i/ka*. This optionality tends to arise when the speaker perceives themselves as familiar with the referent, or in registers characterized by neutral description, such as news articles.

While language model performance on syntactic dependencies, including filler-gap dependencies (Wilcox et al., 2018), subject-verb agreement (Linzen et al., 2016; Goldberg, 2019; Guarasci et al., 2021), anaphor binding (Hu et al., 2020), and control phenomena (Lee and Schuster, 2022), has been extensively explored in recent years, comparatively little attention has been paid to non-syntactic dependencies that encode pragmatic constraints such as honorifics. Because both morphosyntax

[1] Abbreviations used in this paper are provided in the appendix.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 369–378
July 3-4, 2026 ©2026 Association for Computational Linguistics

and socio-pragmatic factors condition honorific use, it provides a uniquely revealing testing ground for evaluating whether language models track not only structural dependencies but also the social knowledge that governs their use. The present study addresses this gap by extending Noh et al. (2024). Specifically, whereas their study focused on the presence of the verbal honorific marker *-si*, we employ a full paradigm that also includes the nominative honorific marker *-kkeyse*. Second, while their analyses were limited to Korean-focused encoder models, we also examine multilingual and decoder-based models, enabling a broader assessment of how model architecture and training regime affect the processing of Korean honorifics.

2 Background

2.1 Korean honorific system

There are two main types of honorifics in Korean. First, addressee (or hearer) honorifics indicate the relationship between the speaker and the addressee. When the addressee is socially higher than the speaker, the speaker employs certain sentential endings (e.g., *-yo* or *-supnita*) to express the speaker's deference toward the addressee. On the other hand, subject honorifics reflect the relationship between the speaker and the subject of the sentence. When the subject is an entity considered to be honorable by the speaker, special morphological inflections are attached to the subject (i.e., *-kkeyse*) and the verb (i.e., *-si*), as shown above in (1b).

2.2 Subject honorifics: Human judgment

Unlike honorable subjects, plain subjects do not trigger special morphological inflections. Examples in (2) show that the plain subject *ai* 'child' is not compatible with any honorific markings, where * indicates unacceptability.

(2) a. Ai-ka pyenci-lul ssu-ess-ta.
child-NOM letter-ACC write-PST-DECL

b. *Ai-ka pyenci-lul
child-NOM letter-ACC
ssu-si-ess-ta.
write-HON-PST-DECL

c. *Ai-kkeyse pyenci-lul
child-NOM.HON letter-ACC
ssu-ess-ta.
write-PST-DECL

d. *Ai-kkeyse pyenci-lul
child-NOM.HON letter-ACC
ssu-si-ess-ta.
write-HON-PST-DECL

'The child wrote a letter.'

Examples in (3) show a full paradigm of subject honorifics with an honorable subject *kyoswunim* 'professor'. Interestingly, native speakers found these sentences acceptable even in the absence of honorific markings (Song et al., 2019).

(3) a. Kyoswunim-i pyenci-lul
professor-NOM letter-ACC
ssu-ess-ta.
write-PST-DECL

b. Kyoswunim-i pyenci-lul
professor-NOM.HON letter-ACC
ssu-si-ess-ta.
write-HON-PST-DECL

c. ?Kyoswunim-kkeyse pyenci-lul
professor-NOM.HON letter-ACC
ssu-ess-ta.
write-PST-DECL

d. Kyoswunim-kkeyse pyenci-lul
professor-NOM.HON letter-ACC
ssu-si-ess-ta.
write-HON-PST-DECL

'The professor wrote a letter.'

(3d) illustrates the canonical co-occurrence of both honorific markings *-kkeyse* and *-si*, representing the most expected honorific configuration. However, as shown in (3a), the sentence with an honorable subject without any honorific markings is considered acceptable to native speakers of Korean. Even more intriguingly, sentences are judged acceptable when the verbal honorific marker *-si* is present without *-kkeyse*, as in (3b). The reverse configuration, in which *-kkeyse* appears without *-si* as in (3c), is the least favorable but still considered acceptable, suggesting that native speakers exhibit considerable flexibility in honorific marking when the subject referent is honorable. The marginal acceptability is indicated by ? in the examples.

2.3 Honorifics and language models

Honorifics in language models have not yet been widely studied. Existing work has primarily focused on how the inclusion of honorifics affects model performance on downstream tasks, including text similarity task (Hwang et al., 2024), request speech act accuracy and friendliness (Jung et al., 2024), and pragmatic competence in request expressions (Chen et al., 2025).

This focus on task performance, however, leaves open the question of how models process the honorific system itself. To our knowledge, only two

Condition	Sentence					
a. HON-HON	Ecey	cenyek-ey	honcase	**kyoswunim**-i	pyenci-lul	ssu-**si**-ess-ta.
	yesterday	evening-LOC	alone	**professor**-NOM	letter-ACC	write-**HON**-PST-DECL
	'The professor wrote a letter alone yesterday evening.'					
b. HON-PLAIN	Ecey	cenyek-ey	honcase	**kyoswunim**-i	pyenci-lul	ssu-ess-ta.
	yesterday	evening-LOC	alone	**professor**-NOM	letter-ACC	write-PST-DECL
	'The professor wrote a letter alone yesterday evening.'					
c. PLAIN-PLAIN	Ecey	cenyek-ey	honcase	**ai**-ka	pyenci-lul	ssu-ess-ta.
	yesterday	evening-LOC	alone	**child**-NOM	letter-ACC	write-PST-DECL
	'The child wrote a letter alone yesterday evening.'					
d. PLAIN-HON	*Ecey	cenyek-ey	honcase	**ai**-ka	pyenci-lul	ssu-**si**-ess-ta.
	yesterday	evening-LOC	alone	**child**-NOM	letter-ACC	write-**HON**-PST-DECL
	'The child wrote a letter alone yesterday evening.'					

Table 1: Examples of Experiment 1 main conditions in CLOSE distance (where * indicates unacceptability)

studies have specifically investigated the Korean honorific system in language models from a linguistic perspective. On the one hand, Lee and Wang (2023) show that language models such as GPT-2, RoBERTa, BERT, and ALBERT fail to fully capture addressee honorifics in Korean. On the other hand, Noh et al. (2024) examine whether four Korean BERT-based models including KR-BERT and KoELECTRA process honorific mismatches in a human-like manner. They specifically focus on the presence of the verbal honorific marker *-si*, comparing the patterns in (2a)-(2b) against their honorable counterparts in (3a)-(3b). Their results demonstrate that Korean BERT models exhibit human-like sensitivity to honorific mismatches, suggesting that language models are capable of capturing certain morphosyntactic properties of the honorific system.

Our study investigates whether language models process subject honorifics in a human-like manner, building on Noh et al. (2024). We extend their work in two important aspects. First, whereas their study focused primarily on the presence of the verbal honorific marker *-si*, we employ a full dataset paradigm that also includes the nominative honorific marker *-kkeyse*, allowing for a more comprehensive and complete evaluation of subject honorifics. Second, while their analysis was limited to Korean-trained encoder models, we additionally examine multilingual and decoder-based models, thereby broadening the empirical coverage and enabling a more systematic assessment of how different model architectures and training regimes handle Korean honorifics.

3 Methods

We conducted three experiments to test whether language models process Korean subject honorifics similarly to humans.[2] The experiments build progressively: Experiment 1 establishes a baseline by replicating the experimental design of Noh et al. (2024), examining how verbal honorific marking *-si* interacts with subject honorability, without considering *-kkeyse*. Experiments 2 and 3 then examine the full subject honorifics paradigm by including both *-kkeyse* and *-si*. Importantly, Experiment 2 tests honorific patterns with honorable subjects, while Experiment 3 uses plain subjects. These design choices allow us to distinguish whether models are sensitive to surface dependency alone (i.e., co-occurrence of *-kkeyse* and *-si*), or whether they additionally encode the pragmatic constraints (i.e., honorability of the referent) that trigger subject honorifics.

3.1 Experiment 1

3.1.1 Stimuli

We manipulated whether the subject was honorable or plain, and whether the verb carried honorific marking *-si*. This created the four experimental conditions detailed in Table 1: HON-HON (honorable subject with honorific verb), HON-PLAIN (honorable subject with plain verb), PLAIN-PLAIN (plain subject with plain verb), and PLAIN-HON (plain subject with honorific verb). We used 10 honorable subjects, 10 plain subjects, and 10 verb pairs, each appearing with and without honorific marking *-si*. In addition, the distance between the subject and verb was manipulated by inserting a three-word adverbial phrase (e.g., *alone yesterday evening*), either between the subject and verb (FAR condition) or before the subject (CLOSE condition). This manipulation tested whether intervening material disrupts the processing of subject honorific

[2]The experimental stimuli and code used in this study are available at https://github.com/chloedkkwon/honorific-dependency.

Condition	Sentence
a. NO HONORIFICS	Kyoswunim-**i** ecey cenyek-ey honcase pyenci-lul ssu-ess-ta. professor-**NOM** yesterday evening-LOC alone letter-ACC write-PST-DECL 'The professor wrote a letter alone yesterday evening.'
b. VERB-ONLY	Kyoswunim-**i** ecey cenyek-ey honcase pyenci-lul ssu-**si**-ess-ta. professor-**NOM** yesterday evening-LOC alone letter-ACC write-**HON**-PST-DECL
c. SUBJECT-ONLY	?Kyoswunim-**kkeyse** ecey cenyek-ey honcase pyenci-lul ssu-ess-ta. professor-**NOM.HON** yesterday evening-LOC alone letter-ACC write-PST-DECL
d. ALL HONORIFICS	Kyoswunim-**kkeyse** ecey cenyek-ey honcase pyenci-lul ssu-**si**-ess-ta. professor-**NOM.HON** yesterday evening-LOC alone letter-ACC write-**HON**-PST-DECL 'The professor wrote a letter alone yesterday evening.'

Table 2: Examples of Experiment 2 main conditions in FAR distance (where ? indicates the least favorable option).

dependency patterns (e.g., Lee and Wang, 2023). The distance manipulation doubled the number of test sentences, yielding 800 sentences in total for Experiment 1 (4 conditions × 10 subjects × 10 verbs × 2 distances).

3.1.2 Measurement

We computed surprisal as a proxy for the acceptability of sentences, following proposals that it serves as a computational analog to processing difficulty in human reading (Hale, 2001; Levy, 2008). Under this framework, tokens that are less expected by the language model result in higher surprisal, reflecting the greater cognitive effort required when humans encounter unexpected input. In our study, surprisal was measured at the *verb* because it is the critical region for subject-verb honorific dependency. Thus, surprisal values were calculated by averaging across all tokens comprising the verb.

To calculate surprisal, we first obtained the probability distribution for each token position in the sentence. For masked language models, subword tokens of the target verb were masked individually to predict their probability. In causal language models, the distribution was based on all preceding tokens. Surprisal was then determined by taking the negative log probability of each subword token and averaging them across the entire verb.

$$Surprisal = -\log_2 P(\text{token}|\text{context}) \quad (1)$$

3.2 Experiments 2 and 3

3.2.1 Stimuli

Both Experiments 2 and 3 extended the design of Experiment 1 by adding *-kkeyse*, which occurs with honorable subjects in Korean. First, Experiment 2 used only honorable subjects, resulting in four conditions shown in Table 2: NO HONORIFICS (honorable subject without *-kkeyse* + plain verb), VERB-ONLY HONORIFICS (honorable subject without *-kkeyse* + honorific-marked verb), SUBJECT-ONLY HONORIFICS (honorable subject with *-kkeyse* + plain verb), and ALL HONORIFICS (honorable subject with honorific nominative marker *-kkeyse* + honorific-marked verb). We used the same 10 honorable subjects from Experiment 1 with or without *-kkeyse* and 10 verb pairs with or without the *-si* honorific marking.

Experiment 3 tested the same four conditions but with plain subjects (e.g., *ai* 'child'). In this case, NO HONORIFICS is the only acceptable condition. See Table 3 for example stimuli. We used 10 plain subjects with or without *-kkeyse* and the same 10 verb pairs. Both experiments included the same distance manipulation as Experiment 1, with a three-word adverbial phrase inserted either between the subject and verb (FAR condition) or before the subject (CLOSE condition). We generated 800 sentences (4 conditions × 10 subjects × 10 verbs × 2 distances) for each experiment.

3.2.2 Measurement

We calculated the surprisal of the verb as described in Experiment 1. In addition, as one of the reviewers pointed out, the mismatch between a plain subject with honorific marking in Experiment 3 may be most directly reflected in the subject region itself, rather than at the verb. To address this, we additionally calculated surprisal at the subject region (the subject noun and the nominative honorific marker *-kkeyse*) for Experiment 3, given that models consistently tokenized the subject noun and *-kkeyse* as a single unit.

For Experiment 2 and 3, we also calculated the accuracy, as the four conditions within each item set differ only in honorific marking, making surprisal most directly comparable within an item set.[3] Accuracy operationalizes this by measuring

[3]Note that accuracy is not calculated for Experiment 1, unlike Experiment 2 and 3, because the comparison between conditions is confounded by lexical differences between plain and honorable subjects (e.g., child vs. professor).

Condition	Sentence
a. ALL HONORIFICS	*Ai-**kkeyse** ecey cenyek-ey honcase pyenci-lul ssu-**si**-ess-ta. child-**NOM.HON** yesterday evening-LOC alone letter-ACC write-**HON**-PST-DECL
b. VERB-ONLY	*Ai-**ka** ecey cenyek-ey honcase pyenci-lul ssu-**si**-ess-ta. child-**NOM** yesterday evening-LOC alone letter-ACC write-**HON**-PST-DECL
c. SUBJECT-ONLY	*Ai-**kkeyse** ecey cenyek-ey honcase pyenci-lul ssu-ess-ta. child-**NOM.HON** yesterday evening-LOC alone letter-ACC write-PST-DECL
d. NO HONORIFICS	Ai-**ka** ecey cenyek-ey honcase pyenci-lul ssu-ess-ta. child-**NOM** yesterday evening-LOC alone letter-ACC write-PST-DECL 'The child wrote a letter alone yesterday evening.'

Table 3: Examples of Experiment 3 main conditions in FAR distance (where * indicates unacceptability).

whether the model's ranking of conditions within each item set aligns with human judgment. Thus, for Experiment 2, we awarded one point per item set if the model correctly assigned the highest surprisal to the SUBJECT-ONLY HONORIFICS condition, which is the least favorable option (marked with ? in Table 2). Accuracy was then computed as the number of points earned divided by the total number of item sets. For Experiment 3, a point was awarded if the models correctly predicted the lowest surprisal for the NONE HONORIFICS condition, which is the only acceptable option with plain subjects. Accuracy was calculated in the same way.

For each experiment, we fitted a linear mixed-effects model (LME) using *lme4* (Bates et al., 2015) in R (R Core Team, 2023), with surprisal at the target verb as the dependent variable and item set number as a random intercept. Model selection via AIC (Akaike, 1974) favored the maximal fixed-effects structure including all two- and three-way interactions between main condition, distance, and language model. We then used the *emmeans* package (Lenth, 2023) to examine differences within each factor of a model.

3.3 Language models

We tested seven language models: two encoder-based and five decoder-based. The encoder-based models were KoBERT (Kim, 2020) and KoELECTRA (Park, 2020), both based on BERT (Devlin et al., 2019) and ELECTRA (Clark et al., 2020) architectures, respectively, both with ~110M parameters. Although these models are smaller than the decoder-based models, no Korean encoder-based models with comparable (~1B) parameter counts were publicly available at the time of this study. Both models have been widely used in previous Korean NLP studies (Hur et al., 2021; Kim and Jo, 2022; Hwang and Oh, 2024), making them appropriate baselines.

The decoder-based models included three

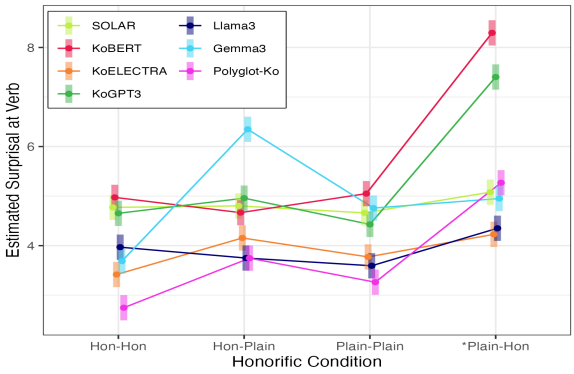


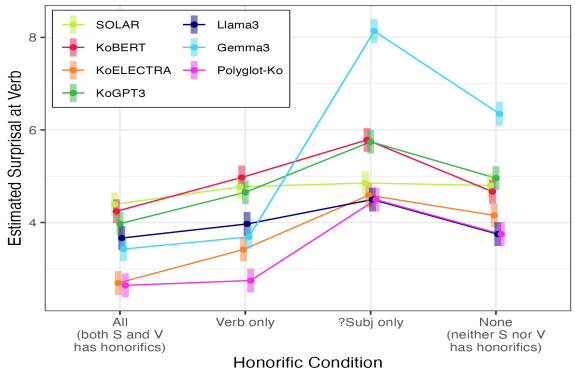
Figure 1: Estimated target verb surprisal across conditions in Experiment 1.

Korean-focused models: Polyglot-Ko (Ko et al., 2023) (1.3B parameters), trained exclusively on 863GB of Korean text; KoGPT 3 (SK Telecom, 2021) (1.2B), trained on a proprietary Korean corpus; and SOLAR (Kim et al., 2023) (10.7B), trained on a Korean- and English-focused multilingual corpus. We also included two multilingual models, Gemma 3 (Gemma Team, 2025) (1B) and Llama 3 (Grattafiori et al., 2024) (1B), trained on large multilingual datasets where Korean is present but not a primary focus. This allows us to compare Korean-focused models against multilingual models. Note that SOLAR is considerably larger than the others, as no smaller variant was publicly available.

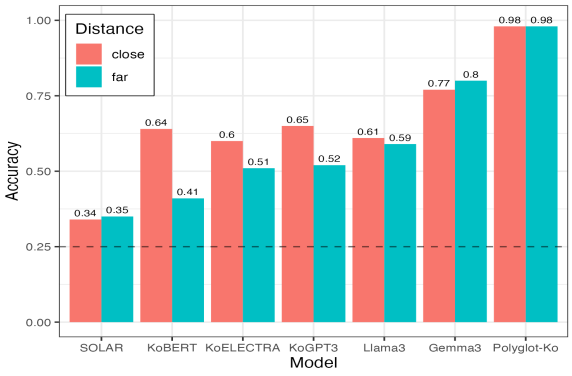
4 Results

4.1 Experiment 1

In Experiment 1, we compared surprisal across conditions that combine plain or honorable subjects and verbs with or without honorific marking *-si*. The results are presented in Figure 1, with colors representing each model. Most models assigned the highest surprisal to the unacceptable PLAIN-HON condition than to other conditions, except for Gemma 3 (light blue). This pattern is consistent



(a) Estimated target verb surprisal



(b) Model performance accuracy

Figure 2: Results of Experiment 2. In (b), the horizontal line indicates the chance level.

with Noh et al. (2024), although our results are different in that their HON-HON condition yielded higher surprisal than the HON-PLAIN condition, whereas we observed this pattern only in Gemma 3, and to a lesser extent, in KoELECTRA (orange). In this model, surprisals in the HON-PLAIN condition exhibited a wide distribution with a lower median than PLAIN-HON, rather than clearly elevated surprisal. In addition, the FAR condition showed slightly lower surprisal than the CLOSE condition for all models (mean difference: 0.49).

4.2 Experiment 2

In Experiment 2, we used honorable subjects that occur with or without *-kkeyse* and verbs with or without *-si*. The results are shown in Figure 2a. All models except SOLAR (yellow green) exhibited the highest surprisal in the SUBJECT-ONLY condition, where an honorable subject occurs with the subject honorific marking *-kkeyse* but with a plain verb (i.e., without *-si*), making the sentence least favorable by human subjects (Song et al., 2019).

Figure 2b shows the accuracy of each model by the distance condition, ordered by the model's overall accuracy. It shows that all models achieve beyond chance level accuracy (0.25). Note that surprisal results in Figure 2a and accuracy results in Figure 2b may not necessarily align, because surprisal values are averaged across items within each condition while accuracy is calculated per item set. For instance, KoELECTRA (orange) shows a left-to-right increase in surprisal consistent with human judgments, yet this does not entail high accuracy, as condition-level averaging captures the overall trend across items rather than performance on each individual item.

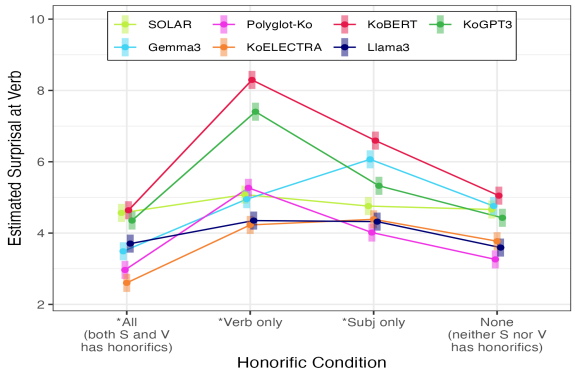
The Korean-focused decoder model Polyglot-Ko showed the highest accuracy, followed by the multilingual decoder model Gemma 3. Models' error patterns were distinctive: Polyglot-Ko nearly always predicted the highest surprisal for the SUBJ-ONLY condition, but its incorrect predictions (2%) consistently assigned the highest surprisal to the ALL condition, which is the most favorable option for human subjects. Gemma 3 exhibited a different pattern, usually assigning the highest surprisal to the NONE condition when wrong. These patterns contrast with all other models, which assigned the highest surprisal to the VERB-ONLY condition when wrong.

Regarding the distance condition, most models performed better in the CLOSE condition, likely reflecting sensitivity to subject-verb adjacency, while Polyglot-Ko and Gemma 3 showed similar accuracy across both conditions. Surprisal in the FAR condition was consistently lower than in the CLOSE condition for most models across all experiments: an intervening adverbial phrase between the subject and verb lowers surprisal across the board.

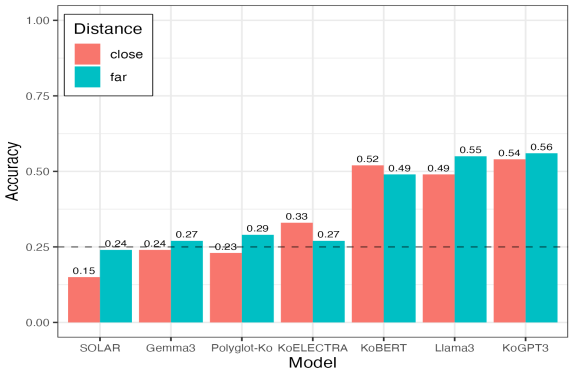
4.3 Experiment 3

The results from Experiment 3 demonstrate that most models generally exhibit high surprisal for the VERB-ONLY and SUBJECT-ONLY conditions at the verb region, as shown in Figure 3a. This aligns with human judgments that plain subjects such as *ai* 'child' are not expected to take honorific markings. However, the ALL condition, where both subject and verb are honorific-marked, yielded similar or even lower surprisal than the NONE condition in all models, despite the latter condition being the only acceptable option for native speakers.

Figure 3b presents model accuracy for Experiment 3. Again, a point was awarded if the mod-



(a) Estimated target verb surprisal



(b) Model performance accuracy

Figure 3: Results of Experiment 3 at the verb region. In (b), the horizontal line indicates chance level.

els predicted the lowest surprisal for the NONE condition since plain subjects are not expected to co-occur with any honorific marking. Overall accuracy was lower than in Experiment 2. More than half of the models achieve around or below the chance level accuracy (0.25). KoGPT 3 achieves the best performance by correctly predicting the lowest surprisal for the NONE condition in 55% of items across distance conditions. Unlike Experiment 2, error patterns were consistent: all models preferentially predicted the ALL condition as having the lowest surprisal. Note that the effect of distance varied by model type: encoder-based models (KoELECTRA and KoBERT) performed better in the CLOSE condition while decoder-based models performed better in the FAR condition, to varying degrees.

Figure 4 shows surprisals of each model at the subject region. KoELECTRA and decoder-based models, except for SOLAR, exhibited higher surprisals when plain subjects occurred with the honorific nominative marker, aligned with human judgments. Specifically, the ALL condition yielded higher surprisal than VERB-ONLY, and the SUBJECT-ONLY condition yielded higher surprisal than NONE condition. In contrast, KoBERT (red) showed the opposite pattern when comparing the ALL and VERB-ONLY conditions, yielding higher surprisal for VERB-ONLY than for ALL.

5 Discussion

This study examined whether seven language models have acquired knowledge of Korean subject honorifics, using surprisal as a proxy for acceptability across three experiments. The results reveal a systematic dissociation: models were generally good at detecting surface morphosyntactic

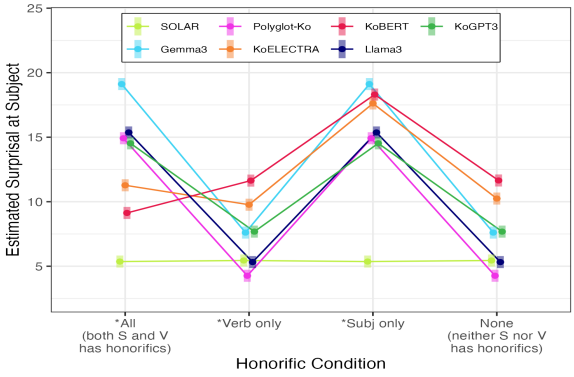


Figure 4: Estimated subject surprisal across conditions in Experiment 3.

mismatches but failed to respect the pragmatic constraints that govern honorific use. We discuss these findings in terms of what they reveal about the nature of linguistic knowledge in language models and broader implications for our understanding of language models' capabilities and limitations.

5.1 Surface heuristics vs. socio-pragmatic grounding

Experiments 1 and 2 demonstrated that most models assigned higher surprisal to less favorable honorific constructions (i.e., the PLAIN-HON condition in Experiment 1 and the SUBJECT-ONLY condition in Experiment 2). However, Experiment 3 revealed critical limitations. Most models, except SOLAR and KoBERT, correctly assigned higher surprisal to the subject region when plain subjects occurred with the honorific nominative marker, suggesting some sensitivity to the mismatch between the status of the subject and honorific marking at the subject position. However, models except Llama 3 incorrectly assigned the lowest surprisal to the verb

region when plain subjects occurred with both *-kkeyse* and *-si* (i.e., ALL condition), despite this condition being unacceptable to native speakers. Rather than treating the NONE condition as the most expected, as human speakers would, models appeared to generalize a surface-level preference for co-occurrence between subject nominative marker *-kkeyse* and verb honorific marker *-si*, irrespective of whether the subject referent warrants honorific treatment. These findings suggest that models have learned a surface co-occurrence heuristic between subject and verb honorific markers, instead of internalizing the underlying socio-pragmatic principle that honorific marking is licensed by the social status of the referent. In other words, models appear to prioritize surface distributional patterns over pragmatic appropriateness. Our results indicate that even Korean-focused models, which have extensive exposure to Korean honorific patterns, have not fully acquired this socio-pragmatic dimension.

These findings align with broader arguments in the literature that language models excel at learning distributional patterns but struggle with context-dependent reasoning that requires grounding in social knowledge (e.g., Sap et al., 2022). Korean honorifics present a particularly interesting test case for this distinction because the acceptability of a construction depends not just on morphosyntactic well-formedness but on a generalized understanding of the socio-pragmatic conditions under which honorific marking is licensed. This can be further illustrated by two types of cases that are difficult to account for on the basis of distributional patterns alone. First, honorific marking can occasionally be used with non-honorable referents such as pets, when the speaker playfully elevates the social status her pet (e.g., *Wuli koyangi-**kkeyse** pappu-**si**-ta.* roughly tranlated as 'His majesty, my cat, is busy.'). Second, even with an inherently honorable referent, a speaker may opt out of honorific marking entirely when she perceives herself as close and familiar with that referent (see also Song et al., 2019). Future work should construct experiments that directly manipulate such factors, which would allow us to probe whether language models can capture deeper level of pragmatic knowledge.

5.2 Distance sensitivity

A secondary finding across all three experiments was that the distance between subject and verb modulated model performance. Encoder-based models (KoBERT and KoELECTRA) consistently performed better in the CLOSE condition across all three experiments, while decoder-based models showed more variable patterns.

Interestingly, models generally showed lower surprisal in the FAR condition despite lower accuracy, presenting a dissociation between these two measures. One possibility is that lower surprisal reflects reduced sensitivity to the honorific dependency between the subject and verb. For instance, because intervening material dilutes the local context, the models may end up assigning less extreme surprisal values across all conditions, thereby reducing their ability to find unacceptable constructions more surprising. However, we could not conclude from the current data that the two patterns are directly related.

5.3 Korean-focused vs. multilingual models

Contrary to our hypothesis that Korean-focused models would consistently outperform multilingual models, the results did not show a clear advantage for Korean-focused training. For example, Llama 3, a multilingual model, achieved the second-highest accuracy in Experiment 3 and performed comparably to Korean-focused models. Meanwhile, SOLAR, despite being both Korean-focused and considerably larger than all other models, did not consistently outperform either multilingual or smaller models. These findings echo broader observations in the multilingual NLP literature that the relationship between language-specific training data and downstream linguistic performance is not straightforwardly linear (Conneau et al., 2020).

5.4 Limitations

One limitation of the present study is the absence of human judgment data for Experiment 3. Although it is a well-established fact in the literature that plain subjects do not trigger any honorific markings (e.g., Kim and Sells, 2007; Choi and Harley, 2019), empirical validation would strengthen the conclusions.

References

Hirotugu Akaike. 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

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.

Yijun Chen, Peng Yue, and Henry Davidge. 2025. Comparing generative AI with native speakers in terms of request expressions in Japanese. *Discover Education*, 4(478).

Jaehoon Choi and Heidi Harley. 2019. Locality domains and morphological rules: Phases, heads, node-sprouting and suppletion in Korean honorification. *Natural Language & Linguistic Theory*, 37(4):1319–1365.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proceedings of the 8th International Conference on Learning Representations*.

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.

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*, pages 4171–4186. Association for Computational Linguistics.

Gemma Team. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Raffaele Guarasci, Stefano Silvestri, Giuseppe De Pietro, Hamido Fujita, and Massimo Esposito. 2021. Assessing BERT's ability to learn Italian syntax: a study on null-subject and agreement phenomena. *Journal of Ambient Intelligence and Humanized Computing*, 14.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 159–166. Association for Computational Linguistics.

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.

Yuna Hur, Suhyune Son Son, Midan Shim, Jungwoo Lim, and Heuiseok Lim. 2021. K-EPIC: Entity-perceived context representation in Korean relation extraction. *Applied Sciences*, 11(23):11472.

Sunik Hwang and Hayoung Oh. 2024. Trust beyond numbers: Data augmentation formula for poll prediction. *KSII Transactions on Internet and Information Systems*.

Yerin Hwang, Yongil Kim, Jeesoo Bang, Hyunkyung Bae, Hwanhee Lee, and Kyomin Jung. 2024. Kosmic: Korean text similarity metric reflecting honorific distinctions. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9954–9960, Torino, Italia. ELRA and ICCL.

Gayeon Jung, Joeun Kang, Fei Li, and Hansaem Kim. 2024. Are large language models affected by politeness? Focusing on request speech acts in Korean. In *Proceedings of the 38th Pacific Asia Conference on Language, Information and Computation (PACLIC 38)*, pages 1–10.

Dahyun Kim, Chanjun Park, Sanghoon Kim, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Jung, and 1 others. 2023. SOLAR 10.7b: Scaling large language models with simple yet effective depth up-scaling. *Preprint*, arXiv:2312.15166.

Gyeongmin Kim and Jaechoon Jo. 2022. Verification of a dataset for Korean machine reading comprehension with numerical discrete reasoning over paragraphs. *JOIV: International Journal on Informatics Visualization*, 6(2-2):587–592.

Jong-Bok Kim and Peter Sells. 2007. Korean honorification: a kind of expressive meaning. *Journal of East Asian Linguistics*, 16(4):303–336.

Sungjoon Kim. 2020. KoBERT. GitHub repository.

Hyunwoong Ko, Kichang Yang, Minho Ryu, Taekyoon Choi, Seungmu Yang, jiwung Hyun, and Sungho Park. 2023. A technical report for Polyglot-Ko: Open-source large-scale Korean language models. *Preprint*, arXiv:2306.02254.

Soo-Hwan Lee and Sebastian Schuster. 2022. Can language models capture syntactic associations without surface cues? A case study of reflexive anaphor licensing in English control constructions. *Proceedings of the Society for Computation in Linguistics*, 5(1):206–211.

Soo-Hwan Lee and Shaonan Wang. 2023. Do language models know how to be polite? *Society for Computation in Linguistics*, 6(1):375–378.

Russell V. Lenth. 2023. *emmeans: Estimated Marginal Means, aka Least-Squares Means*. R package version 1.8.9.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

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.

Kangsan Noh, Sanghoun Song, and Eunjeong Oh. 2024. How language models understand honorific mismatches in korean. *Language Research*, 60(3):303–322.

Jangwon Park. 2020. KoELECTRA: Pretrained ELECTRA model for Korean. GitHub repository.

R Core Team. 2023. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Maarten Sap, Ronan Le Bras, Daniel Fried, and Yejin Choi. 2022. Neural theory-of-mind? on the limits of social intelligence in large LMs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3762–3780, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

SK Telecom. 2021. Ko-GPT-Trinity 1.2b. Hugging Face model repository.

Sanghoun Song, Jae-Woong Choe, and Eunjeong Oh. 2019. An empirical study of honorific mismatches in Korean. *Language Sciences*, 75:47–71.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.

A Abbreviations

ACC = Accusative case; DECL = Declarative sentential ending; HON = Verbal honorific morpheme; LOC = Locative postposition; NOM = Nominative case; NOM.HON = Honorific nominative case; PST = Past tense.

CrosSing: Cross-Scale Reasoning Evaluation on LLMs against Humans

Qi Han[*]
Cornell University
Linguistics Department
qh246@cornell.edu

Yifan Wu[*]
Cornell University
Linguistics Department
yw2578@cornell.edu

Marten van Schijndel
Cornell University
Linguistics Department
mv443@cornell.edu

Abstract

While many studies have shown LLMs perform well in various reasoning tasks, few have examined their capacity on semantic reasoning tasks. As LLMs reason with language, it is crucial to understand how well they grasp and use the underlying scalar relationships in language. In this study, we introduced a new dataset **CrosSing** (Cross-Scale reasoning), providing a human baseline against which to evaluate LLMs' ability to reason across lexical scales in gradable adjectives. We further probed how their understanding is influenced by overinformative contexts. We evaluated ten high-performing LLMs and found that some outperformed humans when no extra information was provided, but that LLM performance declined in certain overinformative contexts while human performance improved significantly. This contrast reveals a fundamental difference between recent LLMs and humans in understanding adjectives' scalar relationships and how such understanding behaves in overinformative contexts.

1 Introduction

Opposing scales are common in real life use cases. Consider the following example shown in Figure 1, a chocolate seller categorizes his chocolates into four specific tiers: free, cheap, expensive, and luxurious, and he asks an LLM-based chatbot for pricing strategies based on the tiers. To give a reasonable response, the chatbot needs to rank the four tiers on a unified scale to map their relative orderings to appropriate price values.

Tasks like this chocolate-pricing example rely on the ability to map and order adjectives from multiple scales onto a single shared dimension. This can occur with abstract properties such as affordability and concrete scalar values like prices. Such an ordering usually involves an understanding of scalar implicatures (SI; Grice, 1975; Horn, 1989; Levinson, 2000, a.o.), where listeners draw an inference that a speaker's use of a weaker expression implies the negation of a stronger alternative (e.g., *expensive* → "*expensive but not luxurious*"). While human SI reasoning is well-explored (e.g., Noveck, 2001; Papafragou and Musolino, 2003; Bott and Noveck, 2004; Pouscoulous et al., 2007; Katsos and Bishop, 2011; Frank and Goodman, 2012; Goodman and Stuhlmüller, 2013; Degen, 2015; Degen and Tanenhaus, 2015, 2016; Goodman and Frank, 2016; Skordos and Papafragou, 2016; Gotzner et al., 2018; Kampa et al., 2025), less is known about whether large language models (LLMs) align with humans in representing and reasoning with adjectives across opposing scales.

Figure 1: Example of cross-scale reasoning. Picture generated by Nano Banana Pro.

Recent computational work has shown that language models can induce scalar structures and encode SIs in a parallel fashion to human pragmatic reasoning (Kim and de Marneffe, 2013; Schuster et al., 2020; Soler and Apidianaki, 2020; Hu et al., 2023; Nizamani et al., 2024, a.o.). However, previous studies have focused on single-scale SIs. The present study contributes to this line of work by proposing a new task that asks human participants and LLMs to reason **across** scales in different information contexts.

We tested human and LLM **(i)** general understanding of cross-scale adjectival relations (Sec-

[*]Equal contribution.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 379–407
July 3-4, 2026 ©2026 Association for Computational Linguistics

tion 5), and **(ii)** ability to integrate task information (Section 6.1) or semantic information (Section 6.2) in cross-scale reasoning. As LLMs are increasingly trained to perform human tasks, it is crucial to have a parallel LLM-human comparison. LLMs encounter gradable adjectives frequently and are expected to reason about their underlying scales to map words into a coherent meaning space. If model ordering of these fundamental concepts differs from humans, then problems which depend on the accurate usage of these adjectives will naturally emerge. Our results show that while models can order cross-scale adjectives accurately in isolation, their representations are easily disturbed by extra information that caused no harm for, and in some cases aided, human reasoning.

2 Related Work

Gradable adjectives instantiate context-sensitive scales that yield scalar inferences (Beltrama and Xiang, 2013; Ronai and Xiang, 2022, a.o.). Experimental work has shown significant scalar diversity across scales, such that some scales prompt near-categorical implicatures, whereas other scales elicit low SI rates (van Tiel et al., 2016; van Tiel and Schaeken, 2017; van Tiel and Pankratz, 2021; Ronai and Xiang, 2023; Aparicio and Ronai, 2024, 2025). Building on these insights, recent work has explored how LLMs represent and reason over scalar structure.

Researchers have operationalized LM implicature modeling in various ways. Kim and de Marneffe (2013) introduced a distributional method that interpolates between word-embedding vectors to induce adjectival scales. Following Kim and de Marneffe (2013), subsequent methods using neural embeddings aimed to recover known scalar relations from distributional representations, achieving similar results in predicting human-like scalar rankings. For instance, Soler and Apidianaki (2020) leveraged BERT (Devlin et al., 2019) to rank gradable adjectives by intensity, and van Tiel et al. (2019) confirmed the expected scale orderings of adjectives against corpus data. Schuster et al. (2020) trained a neural classifier on human judgment data to predict SIs and showed that some linguistic features can suffice for accurate SI classification. Fine-tuned T5 models' probability estimates and expectation scores for omitted strong alternative adjectives closely mirror human SI variability (Hu et al., 2022, 2023). More recently, SIGA, a large

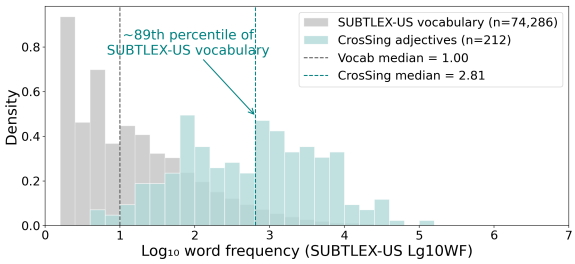


Figure 2: CrosSing Items Frequency Analysis: The median frequency of adjectives in CrosSing is at the 89th percentile of SUBTLX-US vocabulary.

natural language inference (NLI) benchmark for SIs among gradable adjectives by Nizamani et al. (2024), reported that fine-tuning improves detection accuracy of SIs while fine-grained contextual reasoning with gradable adjectives remains challenging. Similarly, broader NLI evaluation benchmarks such as the IMPRES dataset (Jeretic et al., 2020) and the Pragmatics Understanding Benchmark (PUB; Sravanthi et al., 2024) spanning over multiple pragmatic phenomena, demonstrate that fine-tuning LLMs on pragmatic examples markedly boosts performance. Nevertheless, modern LLMs still struggle to integrate subtle context cues across a wide range of pragmatic inferences, not limited to SI (Parrish et al., 2021; Qiu et al., 2023; Ruis et al., 2023; Cho and Kim, 2024, a.o.).

However, these existing studies have focused on how LMs encode and retrieve information within individual lexical scales, whereas no work has systematically examined their ability to pragmatically reason across multiple scales for scale composition.

3 CrosSing

To test how humans and models compose scales from gradable adjectives, we designed the dataset **CrosSing** (Cross-Scale Reasoning Evaluation), where each item consists of four adjectives from two opposing SI scales (e.g., [*free, cheap, expensive, luxurious*]). A total of 109 items were curated with 116 unique SI scales. Among those, 59 single-scale lexical pairs are from Aparicio and Ronai (2025), and another 8 pairs are from Nizamani et al. (2024), to which we added 49 pairs.[1]

The frequency of adjectives used in **CrosSing** ranges from the 28th to 99.9th percentile of the SUBTLEX-US dataset by Brysbaert et al. (2012), whose 74k vocabulary frequency is extracted from

[1]CrosSing: https://github.com/qihan-ling/CrosSing

cheap	free	expensive	luxurious
W_1	W_2	W_3	W_4
Two SI scales		$\langle W_1, W_2\rangle, \langle W_3, W_4\rangle$	
VALID RANKINGS		**4312, 2134**	
INVALID RANKINGS		22 permutations of 1–4	
COMMON WRONGS		1234, 4321, 2143, 3412	

Table 1: Example Item from Dataset and Possible Rankings. The numbers 1 and 3 refer to weak adjectives while 2 and 4 refer to strong ones in opposing SI scales 1-2 and 3-4. These rankings are order-invariant. Common Wrongs are top 4 error types from human experiments.

51 million words of American TV and movie subtitles. The median **CrosSing** adjective is more frequent than 89% of the SUBTLEX-US vocabulary (See Figure 2).

In each SI scale (e.g., [*expensive, luxurious*]), there is a strength comparison between the weak adjective (e.g., *expensive*) and the strong alternative (e.g., *luxurious*). Additionally, the two opposing scales share a unified scale (e.g., *'value'* for the example in Table 1). Items across these opposing single SI scales form antonymous pairs of varying intensity (e.g., [*cheap-expensive*, *free-luxurious*, *cheap-luxurious*, *free-expensive*]).

With four words in an item, there are 24 possible permutations to order them. When the goal is to rank them on the composed scale, there are only **two** satisfactory ranking results as illustrated in Table 1. For any item, the valid rankings must satisfy two conditions: (1) two words in the same SI scale and weak words of opposing SI scales are always direct **neighbors** to each other in the valid rankings (e.g., [*cheap-free, expensive-luxurious, cheap-expensive*]); (2) the weak **antonym** pair always occupies the middle positions (e.g., [*..., cheap , expensive...*]) and the strong pair sits on the two ends (e.g., [*free, ..., ..., luxurious*]). These two rankings for each item are treated as ground truths in the experiments below.[2]

4 Models

We selected ten models from five model families to test via web interfaces: *ChatGPT 4o* (Achiam et al., 2023), *ChatGPT o4-mini-high* (OpenAI, 2025), *Claude 3.7 Sonnet* (Anthropic, 2025), *Gemini 2.0 Flash*, *Gemini 2.5 Pro* (Comanici et al., 2025), *Qwen2.5 Max*, *Qwen3-30B*, *Qwen3-235B* (Yang et al., 2025), *DeepSeek R1* (Guo et al., 2025), *DeepSeek Prover V2* (Ren et al., 2025). These models include both open-sourced and closed-sourced language models. For every zero-shot prompt, we tested it on a separate chat to avoid contamination from other prompts. For ChatGPT models that are sensitive across conversations, we used its temporary chat functionality to disable it from using separate conversations as contexts.

[2]Equivalently, one may describe our analyses as reversal invariant.

5 Just Ranking

We zero-shot evaluated the chosen LLMs on the **CrosSing** task using three ranking conditions that we concurrently used to evaluate human scalar composition in a separate manuscript. Human participants completed a spatial drag-and-drop task, adapted from Starr et al. (2025), requiring them to rank four randomized adjectives along a continuous line, see Figure 3. Both humans and LLMs were evaluated under these three conditions: (1) ranking with no additional information (Just Ranking), (2) ranking with extra task information, and (3) ranking with extra semantic information.

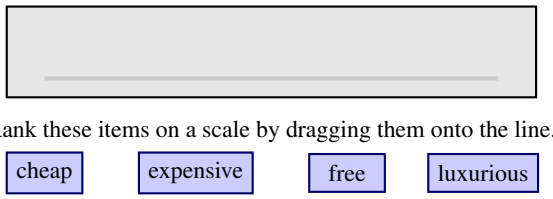


Figure 3: Human study item example, whose valid ordering is demonstrated in Table 1.

5.1 Human Baseline

Our human study established a performance baseline of 76% accuracy at this task, reflecting the extent to which participant responses aligned with the valid rankings, suggesting that human reasoners can map opposing scalar pairs onto a single semantic continuum. We noticed a significant positive correlation between item frequency and accuracy ($\rho = 0.23$ with $p < 0.01$), indicating that less familiar words are more difficult for humans to order. Among the possible 22 invalid rankings, 19 unique permutation types were present in human production. Four dominant wrong rankings (Table 1) simply concatenate the two opposite lexical scales (e.g., [cheap $\rightarrow$ free $\not\rightarrow$ expensive $\rightarrow$ luxurious] or [free $\rightarrow$ cheap $\not\rightarrow$ luxurious $\rightarrow$ expensive]). These dominant wrong rankings preserve the antonym relationships between the two

Model	$Acc\%$	U	ρ	$Acc_N\%$	U_N	ρ_N	$Acc_A\%$	U_A	ρ_A
Qwen3-235B	15.60	23	0.42	35.44↑	25	**0.79**	41.20↑	27	**0.78**
Qwen3-30B	65.75	14	**0.63**	28.81	25	**0.74**	32.19	25	**0.56**
Qwen2.5 Max	69.42	7	**0.56**	59.36	24	**0.79**	54.69	24	**0.63**
ChatGPT 4o	7.65	24	**0.45**	17.94↑	24	**0.76**	18.25↑	24	**0.82**
ChatGPT o4-mini-high	**94.50**	6	**0.8**	**87.67**	27	**0.81**	74.00	10	**0.84**
Gemini 2.0 Flash	**82.87**	9	**0.57**	18.86	24	**0.71**	49.80	24	**0.78**
Gemini 2.5 Pro	**94.8**	6	**0.8**	**95.82**↑	12	0.51	**80.65**	16	**0.79**
Claude 3.7	**78.90**	9	**0.7**	63.61	22	**0.82**	70.23	16	**0.76**
Deepseek-R1	**85.32**	12	0.5	54.57	20	**0.68**	75.03	25	**0.68**
Deepseek-Prover-v2	**88.69**	7	**0.6**	28.20	30	**0.76**	46.84	25	**0.68**
Human (baseline)	**76**	**21**	-	**81**	**19**	-	**76**	**20**	-

Table 2: All Experiments Results: Ten LLMs & human Acc (average accuracy), U (the number of unique ranking responses), and ρ (each model's response pattern's spearman correlation to the human baseline, bold values means $p < 0.01$, and values are highlighted if higher than 0.7), '$_N$' means the metric values are for Extra Task Information experiment, and '$_A$' refers to the Extra Semantic Information experiment.

SI scales and the weak-strong relationships within each subscale, but the composition of the two scales is flawed. We contrast these results with the LLMs results below.

5.2 LLMs Mis-/Super-Alignment

To avoid ordering effects of multi-turn conversations, we provided all 109 experimental items in one prompt (see prompts in Appendix 5). Three randomized orderings of items were run in separate conversations for each model. We found that six out of ten models outperformed humans on accuracy and had a substantially smaller range of wrong answers (Table 2). Models like ChatGPT o4-mini-high and Gemini 2.5 Pro achieved 94% accuracy, 18% higher than the human baseline with 15 fewer wrong answer types. The two models that produced similar numbers of unique ranking responses to humans (Qwen3-235B and ChatGPT4o) had the lowest accuracies (16% and 8%).

While higher-performing models like Gemini 2.5 Pro and ChatGPT o4-mini-high have a smaller range of answer rankings, they showed a high Spearman correlation to humans in terms of answer patterns ($\rho = 0.8$ with $p < 0.001$). Models like Qwen3-30B and Qwen2.5 Max with lower accuracies than humans exhibited a moderate correlation as well. The two worst-performing models (Qwen3-235B and ChatGPT4o) have the lowest correlation with human responses because they assign probability roughly evenly across the full range of rankings, in contrast to a skewed distribution over correct answer types which we see in humans and in other models.

Comparing models of the Qwen and ChatGPT families revealed that increasing model size does not ease the difficulty of the task. We found that smaller models exhibited more human-like response patterns and even had higher performance than humans (Qwen3-30B vs Qwen3-235B, and ChatGPT o4-mini-high vs ChatGPT 4o).

Given that models with better performance have a much smaller answer space than humans, we hypothesize that the training of models may have restricted their output space to the most common patterns that humans tend to produce, resulting in a peakier distribution over possible rankings than humans exhibit.

We observed that the error distribution for six out of ten LLMs have was significantly positively correlated with humans. Though ChatGPT4o has the lowest correlation of $\rho = 0.45$ to humans over all answer types, it shows moderate positive rank correlation with humans over only invalid answers ($\rho = 0.61$). That is, the errors that humans make more often, ChatGPT 4o also makes slightly more often (See Table 9 in Appendix G). In contrast to ChatGPT 4o, Gemini 2.0 Flash has a moderate correlation of $\rho = 0.57$ over all answer types, but its correlation to humans over only invalid answers is insignificant, implying that while it correctly produces more correct answers, it has different preferences over wrong rankings from humans. The correlation analysis indicates that models performing better than humans on this task do not always align with human preferences of possible ranking

Model	JR	R_N	R_A
Qwen3-235B	0.01	0.09	**0.34**
Qwen3-30B	**0.26**	0.13	**0.35**
Qwen2.5 Max	0.07	0.12	0.16
ChatGPT 4o	**-0.25**	**0.27**	0.23
o4-mini-high	**0.26**	0.08	**0.28**
Gemini 2.0 Flash	**0.31**	0.15	0.16
Gemini 2.5 Pro	**0.30**	0.10	0.13
Claude 3.7	0.22	0.04	**0.39**
Deepseek-R1	0.22	0.03	0.18
DS-Prover-v2	0.23	0.14	**0.31**

Table 3: Models Correlation of Item-wise Accuracy to Humans: JR(Just Ranking), R_N(Ranking with Extra Task Info), R_A(Ranking with Extra Semantic Info). Values with $p < 0.01$ significance are highlighted.

types and worse-performing models can still align with humans in their error distributions.

We further find that seven LLMs are positively correlated with humans in terms of item-wise accuracy. We note that having a positive correlation of ranking patterns does not entail a positive correlation of item-wise accuracy, as ChatGPT 4o has 0.61 ranking patterns correlation with $p < 0.01$ significance, but it has a negative Spearman correlation for item-wise accuracy. This means that while ChatGPT 4o makes similar ordering mistakes to humans, it performs worse on items humans find easy to rank and better on items humans find difficult.

While human results show a weak positive correlation between accuracy and word frequency ($\rho = 0.23$ with $p < 0.05$), none of the models exhibited behavior that was significantly correlated with word frequency (see Figure 6 in Appendix C).

6 Ranking with Extra Information

The super-human accuracy of many models in the Just-Ranking experiment might be because they have been explicitly trained to perform scalar inference. In this section, we tested the depth of their understanding by providing additional information which shouldn't alter their behavior. The motivation here is that reasoning across lexical scales require both understanding of the words (semantics of the words) and translating the representation onto a linear order (ordering as a task). With extra information about these two aspects of the task, we explore whether model behavior can be disrupted by these additions. At best, we expect that models could use the information to better encode the needed information. At worst, models can simply ignore the additional information.

6.1 Extra Task Information

To add extra task information to the Just-Ranking experiment, we appended additional information about the direct **neighbor** of two specific adjectives in the form of '*the word* [A] *and the word* [B] *should be placed next to each other*' (e.g., '*A is "cheap" and B is "expensive"* for ranking the list [free, cheap, expensive, luxurious]'). The neighboring information is always consistent with a valid ranking. To account for prompt formatting effects in LLMs, we used three prompts templates, where extra information can either be given once-for-all in a general manner, or repeated after each item with specific references to the words (see prompts in Appendix 5). Each prompt template was run with three random initializations, and we took the average accuracy as results reported below.

6.1.1 Enhanced Human Performance

When exposed to the extra task information, human participants achieved a mean accuracy of 81%, 5% higher than the Just Ranking baseline. The four dominant error rankings remained the same as those in the Just Ranking experiment. The mean correlation between word frequency and accuracy dropped to 0.03 from 0.23 in the Just Ranking experiment (See Figure 7 in Appendix C). Compared with results from Just Ranking, we observed that the lower frequency items' accuracies increase while the top most frequent items' accuracies dropped slightly (See Appendix G). This demonstrates that humans actively used the extra task information to help their ranking performances.

6.1.2 LLM Performance Drop

In contrast to the human improvement, seven out of ten models exhibited significantly degraded accuracy, ranging from -6% to -64% (Table 2).[3] Only ChatGPT o4-mini-high and Gemini 2.5Pro still surpassed human performance. Among the three improved models (Qwen3-235B, ChatGPT 4o, and Gemini 2.5 Pro), Qwen3-235B and ChatGPT 4o still underperformed with average accuracies of 35.44% and 17.94%.

[3]Some models produced incomplete rankings, resulting in more answer types than 24.

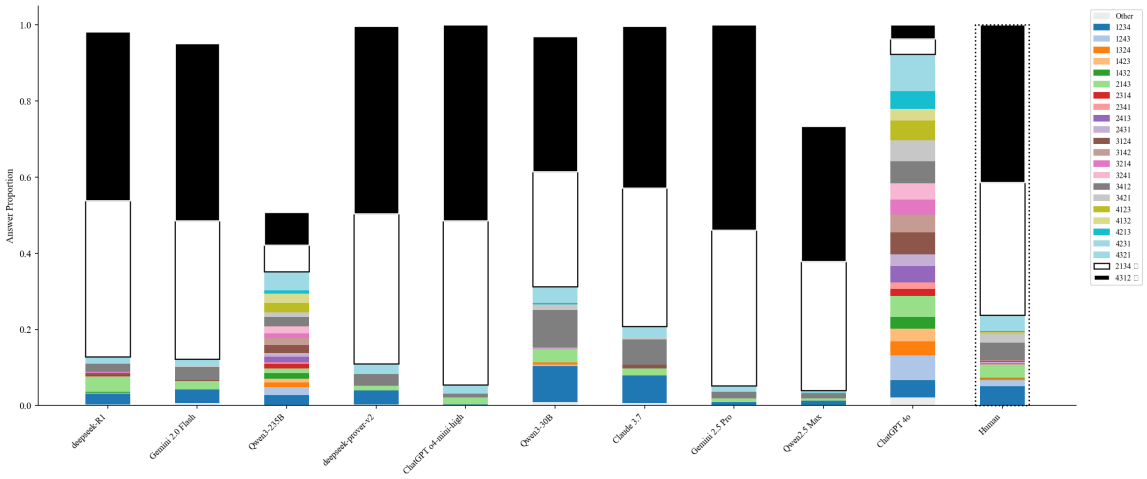


Figure 4: Just Ranking - Response Patterns of LLMs and Humans. Each color bar corresponds to a ranking type. For visual simplicity, low proportion rankings were grouped in the light grey 'other' bar. Black and white bars refer to two order-invariant valid ranking types. NA values (missing responses) are not included, so some models' answer proportions do not add to 1.

All models added at least six new invalid ranking responses over the Just-Ranking experiment. This trend underlines that the extra task information serves more as noise for the models than as guidance for the ranking task. As the extra task information provides partial information to check the correctness of the ranking, it should narrow the range of possible ranking patterns and increase the task accuracy at the same time. It remains a question about why some models (Gemini 2.5 Pro, Qwen3-235B, and ChatGPT 4o) had more kinds of invalid ranking responses but also better accuracies, indicating that, for those models, consideration of more potential rankings co-occurs with higher weights on the correct answers.

Looking closely at model answer distributions (Table 2), we found that all models except Gemini 2.5 Pro have significant strong positive correlation with human ranking patterns ($p < 0.001$), higher than their correlation to humans in the previous Just Ranking experiment. This pattern remained for correlations of error distributions as well (Table 9 in Appendix D). This suggests that the extra task information is noisy instead of informative for models, and when models got disrupted by such noise, they fell back to the data distribution learned from training and generated invalid rankings from the biased human distribution.

Though Gemini 2.5 Pro improved its accuracy by 1% with additional task information, its Spearman correlation of item-wise accuracy to humans decreased to 0.10 from 0.30 in the Just-Ranking experiment (Table 3). This drop in item-wise correlation score means that Gemini 2.5 Pro deviated from its previous agreement on items' difficulty levels to humans. In contrast, ChatGPT 4o's item correlation flipped from negative (-0.25) to positive (0.27), demonstrating a more aligned agreement on item difficulties to humans when given extra task information.

We also find that in this task both Gemini models (2.0 Flash and 2.5 Pro) show significant weak positive correlations between item frequency and accuracy (Figure 7 in Appendix C), while humans and other eight models do not. Gemini 2.5 Pro's correlation score between item frequency and accuracy increased from 0.13 in the Just Ranking experiment to 0.3 ($p < 0.01$) with additional task information, which is the opposite trend we see with humans from 0.23 ($p < 0.05$) in the Just Ranking experiment to 0.03 in current task. This implies that Gemini seems to be more influenced by word frequency than humans in generating rankings with extra task information.

6.1.3 Discussion

ChatGPT o4-mini-high and Gemini 2.5 Pro are the only two models with higher accuracies than humans in this task. Four out of six models that outperformed humans in the Just-Ranking experiment exhibited degraded accuracy to below the human baseline when presented with extra task information about how two words should be put next to each other. Such disruption of those models' behavior indicates that their understanding of

the relations between gradable adjectives is very brittle.

In addition to decreased performance, however, all models except Gemini 2.5 Pro exhibit similar ranking behavior to humans. One possible interpretation for this stronger ranking correlation is that models are initially overtrained or fine-tuned to make basic SI judgments but with additional task information, they fall back on the human data distribution learned during pre-training. However, we also find that the item-wise accuracy correlations with humans are weak and non-significant for all models except ChatGPT 4o. This indicates that models might have learned abstract SI relationships to represent human biases of ranking types beyond memorizing item-wise frequency patterns.

6.2 Extra Semantic Information

In this experiment, we replaced the **neighbor** information in the previous prompt with additional information about the antonymic **semantic** information. This took the form of '*the word* [A] *and the word* [B] *are antonyms*' (e.g., '*A = "cheap" and B = "luxurious" are antonyms*'). Similarly to the Extra Task Information experiment, we ran three randomly initialized runs for each of three prompt templates to calculate the average performances reported below.

6.2.1 Comparable Human Performance

The overall accuracy of 76% and error patterns by humans were comparable to baseline human performance in the Just-Ranking task, suggesting that emphasizing the antonymic relationships between adjectives does not significantly affect human rankings. We found no significant correlation between item frequency and accuracy (Spearman $\rho = 0.07$), though we observe that the extra semantic information boosts the accuracy for the less frequent items more than for the more frequent items with a significance that was merely a trend ($p = 0.07$), likely due to the overall low accuracy gain (see Figure 8 in Appendix C).

6.2.2 LLM Performance Drop

Eight out of ten models exhibited degraded accuracy from -8.66% to -41.84%, including Gemini 2.5 Pro which outperformed humans in the previous two tasks. Even after the drop, Gemini 2.5 Pro still achieved a higher accuracy than humans. The two models which improved in this condition (Qwen3-245B and ChatGPT 4o) were still far below human performance (41.2% and 18.24%).

Similarly to the experiment with extra task information, all ten models expanded their coverage of invalid rankings compared to the baseline Just Ranking experiment, illustrating that the extra information functioned as noise to shift model representations towards considering more types of wrong rankings. We find significant strong positive correlations between models and human for both all-answer and wrong-answer distribution. This shows that despite the distraction, models tend to group words into concatenated single SI pairs, similar to humans.

In contrast to the extra task information experiment, five models exhibited moderate positive item-wise correlation with human accuracy when both humans and models are provided with extra antonymic semantic information (Qwen3-235B, Qwen3-30B, ChatGPT o4-mini-high, Claude 3.7, and Deepseek-Prover-v2). This difference hints that while both kinds of extra information lowered models' performances, the distraction mechanisms diverge for these five models. Namely, extra semantic information causes models to better align item-wise to humans while extra task information does not.

6.2.3 Discussion

We have shown that all models understanding of cross-scale adjectives can be disrupted by extra task or semantic information that does not impede human performances on the same tasks. In both conditions with extra information, 60-80% of tested models exhibited degraded ranking accuracy but their correlation to human ranking patterns greatly increased. When models produce more invalid rankings, they tend to align with the humans bias to simply concatenate the two individual SIs.

While the extra task information reduces the item-wise correlation between humans and most tested models, the extra semantic information improves item-wise correlation with humans for five models. This contrast suggests that models did treat two kinds of additional information as different noises.

We also find that while human accuracy on less frequent items experienced a higher boost with extra information than more frequent items, many model accuracies of more frequent items dropped more than those of less frequent items (See Figure 12 in Appendix C). Such deviation underscores that

these models failed to encode the SIs or use the extra information in a human-like manner and that the disruption effects caused by the extra information were not countered by item frequencies.

6.3 Within-model consistency across ranking conditions

To examine item-level effects from extra information, we compared each model's ranking for each item in the baseline condition to its ranking in the Extra Task / Semantic information condition in terms of validity. We labeled the model's ranking for each item under each condition as valid if the correct ranking occurs in majority among multiple trials under the same experiment condition. We used four types of transitions (*Valid/Invalid→Valid/Invalid*) to categorize the model's performance on tested items. Ten items (∼10% of total items) in the *Valid→Invalid* category for *ChatGPT 4o* under the *Just-Ranking → Extra Task Info* comparison means that *ChatGPT 4o* ranked these items correctly in the baseline but not when extra Task information was provided (see Figure 5).

By comparing items in each transition type of each mode across two extra information conditions, we found that half of the lower-performance models in Just Ranking (*Qwen3-235B, Qwen3-30B, ChatGPT 4o, Gemini 2.0 Flush, Deepseek-Prover-V2*) have high overlap percentages of items across conditions for both *Valid→Invalid* and *Invalid→Invalid* categories, this reinforces that contrary to humans, extra information does not ease the difficulty of ranking for models. The other better-performing half (*Qwen2.5 Max, ChatGPT o4-mini-high, Gemini 2.5 Pro, Deepseek-R1, Claude 3.7*) has high overlap across conditions for both *Invalid→Valid* and *Valid→Valid* (Figure 13 in Appendix H). This pattern demonstrates that on an item-level, models' accuracy scores do not exhibit strong sensitivity to the difference between task and semantic information. In addition, models of the same family have near zero item overlap across transition types, underscoring that model family difference could not sufficiently explain item-level variations among models (see Appendix I).

We also noticed models vary their wrong rankings' choice range by different items. For items that models failed in both Just Ranking and Extra Info conditions (the *Invalid→Invalid* transition), models tended to give different wrong rankings instead of a dominant wrong one (Table 14 in Appendix J). Items in the *Invalid→Valid* transition types under both extra information condition had a clear majority wrong ranking in the baseline, implying for those items the models had strong but incorrect prior beliefs that the extra information helped correct (Figure 17 in Appendix J). For *Valid→Invalid* transitions, most models are more likely to produce a dominant wrong ranking for an item in the Extra Semantic Info condition than the Extra Task Info (Figure 15 in Appendix J), mirroring previous findings that models treat two kinds of extra info as different noises.

6.4 Generation & Judgment (In)Consistency of LLMs

To evaluate the consistency between generation and comprehension, we further compared eight of the tested models.[4] We generated invalid rankings for each item according to common invalid ranking types. In this task, we asked the models to judge whether they agreed with the validity of a given ranking. For each judgment prompt (see example prompt in Table 6 at Appendix B), we mixed valid/invalid rankings of items with varying proportions and asked the models to judge whether they agree with the given rankings. Only one ranking per item is given in a single prompt. We performed 27 trials for each model to calculate the average judgment accuracy. We calculated judgment accuracies by awarding a point whenever a model identifies a valid ranking as correct or recognizes an invalid ranking as incorrect.

We found that not only do all eight models have low accuracies for correct rankings, they also perform worse at the four dominant error patterns of humans than other invalid ranking types (see Figure 11 in Appendix E). The low judgment accuracies for correct rankings echo the finding that model accuracy drops with extra task and semantic information, underscoring the inadequate robustness of cross-scale adjectival understanding. That said, the high correlation of model error patterns with humans also reflects a bias to treat the dominant invalid rankings preferred by humans as more acceptable than other invalid rankings.

7 General Discussion

Among seven models surpassing human performances on the Just-Ranking experiment for cross-

[4]The two missing models were discontinued during the time of this experiment.

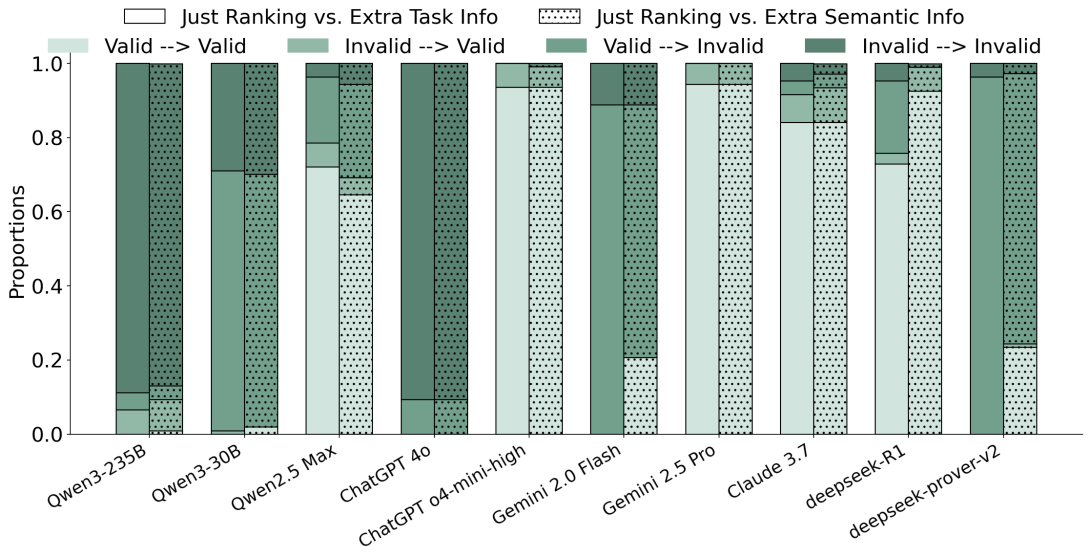


Figure 5: Item-wise ranking validity comparison between the Just Ranking baseline and Extra Task Information (left) / Extra Semantic Information (right). Arrows denote performance changes from the Just Ranking baseline to the extra-information conditions (e.g., Valid (Just Ranking) $\rightarrow$ Valid (Extra Info)).

scale adjectives, only Gemini 2.5 Pro maintained super-human performances on the same ranking tasks with additional information. And even Gemini 2.5 Pro lowered its performance when provided with extra semantic information that is simply a restatement of a lexically encoded antonym relationship between two words.

By comparing models within the same family, we found that bigger model size does not lead to better performance on this task (i.e. Qwen3-235B performed worse than Qwen3-30B). Since all tested models were trained on large amounts of data, they are expected to have high familiarity with items in the CrosSing dataset. Indeed, most models have strong correlations with humans in terms of kinds of produced rankings, biased towards invalid rankings where SI scales are simply concatenated. Such bias is further evidenced in our generation experiment where models commonly fail to recognize dominant invalid human ranking patterns as unacceptable.

Despite exhibiting human-like distributions of ranking patterns, half of the tested models had no significant item-wise correlation with humans in the Just-Ranking experiment and Ranking with Extra Semantic Information experiment. Nine out of ten models had no human-like item-sensitivity for the Ranking with Extra Task Information experiment. This gap in item-sensitivity indicates that these models use their representation of the gradable adjectives differently from humans in the ranking tasks. This finding was further echoed by model insensitivity to item frequency in the ranking experiments; while humans utilized extra information to boost accuracy of low frequency items more than high frequency ones, model performance degraded for high frequency items more than for low frequency items.

8 Conclusion

This study demonstrates that although most models we tested can rank cross-scale adjectives better than humans in a simple setting, their understanding of adjectival relationships is brittle and misaligned with humans. This fragility of model representation is evidenced by big accuracy drops in the presence of additional information that does not interfere with or which actively aids human reasoning. While these models produce human-like distributions of ranking patterns, they do not correlate with item-level human responses.

As the gradable adjectives studied in this paper have median frequency at the 89th percentile of the English vocabulary, if model understanding on these fundamental orderings does not align with humans, then problems dependent on the accurate usage of these adjectives will naturally emerge. This work provides a first step in better understanding model representations of cross-scale reasoning in hopes that future work can investigate methods of better aligning models with human responses.

Limitations

This work only focuses on English gradable adjectives, so the generalizability of models' fragility with extra information remains to be explored. While the gradable adjectives tested in **CrosSing** are of high frequency, the exact training frequency of words remains unknown due to the closed-source nature of the tested models' training data.

References

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

Anthropic. 2025. Claude 3.7 Sonnet System Card. Technical report, Anthropic. Accessed: 2025-10-06.

Helena Aparicio and Eszter Ronai. 2024. Scalar implicature rates vary within and across adjectival scales. *Proceedings of Semantics and Linguistic Theory 33*.

Helena Aparicio and Eszter Ronai. 2025. Scalar implicature rates vary within and across adjectival scales. *Journal of Semantics*, page ffaf002.

Andrea Beltrama and Ming Xiang. 2013. Is 'good' better than 'excellent'? an experimental investigation on scalar implicatures and gradable adjectives. In *Proceedings of sinn und bedeutung*, volume 17, pages 81–98.

Lewis Bott and Ira A. Noveck. 2004. Some utterances are underinformative: The onset and time course of scalar inferences. *Journal of memory and language*, 51(3):437–457.

Marc Brysbaert, Boris New, and Emmanuel Keuleers. 2012. Adding part-of-speech information to the subtlex-us word frequencies. *Behavior research methods*, 44(4):991–997.

Ye-eun Cho and Seong mook Kim. 2024. Pragmatic inference of scalar implicature by llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 10–20.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Judith Degen. 2015. Investigating the distribution of some (but not all) implicatures using corpora and web-based methods. *Semantics and Pragmatics*, 8:11–1.

Judith Degen and Michael K. Tanenhaus. 2015. Processing scalar implicature: A constraint-based approach. *Cognitive science*, 39(4):667–710.

Judith Degen and Michael K. Tanenhaus. 2016. Availability of alternatives and the processing of scalar implicatures: A visual world eye-tracking study. *Cognitive science*, 40(1):172–201.

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.

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

Noah D. Goodman and Michael C. Frank. 2016. Pragmatic language interpretation as probabilistic inference. *Topics in cognitive science*, 20(11):818–829.

Noah D. Goodman and Andreas Stuhlmüller. 2013. Knowledge and implicature: Modeling language understanding as social cognition. *Topics in cognitive science*, 5(1):173–184.

Nicole Gotzner, Stephanie Solt, and Anton Benz. 2018. Scalar diversity, negative strengthening, and adjectival semantics. *Frontiers in psychology*, 9:1659.

Herbert P. Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Laurence R. Horn. 1989. *A natural history of negation*. University of Chicago Press.

Jennifer Hu, Roger Levy, Judith Degen, and Sebastian Schuster. 2023. Expectations over unspoken alternatives predict pragmatic inferences. *Transactions of the Association for Computational Linguistics (TACL)*, 11:885—901.

Jennifer Hu, Roger Levy, and Sebastian Schuster. 2022. Predicting scalar diversity with context-driven uncertainty over alternatives. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, page 68–74.

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.

Alyssa F. Kampa, Anna Papafragou, and Kaja K. Jasińska. 2025. Scalar inference is supported by theory of mind networks in adults and children. *Language, Cognition and Neuroscience*, pages 1–21.

Napoleon Katsos and Dorothy V.M. Bishop. 2011. Pragmatic tolerance: Implications for the acquisition of informativeness and implicature. *Cognition*, 120(1):67–81.

Joo-Kyung Kim and Marie-Catherine de Marneffe. 2013. Deriving adjectival scales from continuous space word representations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1625–1630.

Stephen C. Levinson. 2000. *Presumptive meanings: The theory of generalized conversational implicature*. MIT press.

Rashid Nizamani, Sebastian Schuster, and Vera Demberg. 2024. Siga: A naturalistic nli dataset of english scalar implicatures with gradable adjectives. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14784–14795.

Ira A. Noveck. 2001. When children are more logical than adults: Experimental investigations of scalar implicature. *Cognition*, 78(2):165–188.

OpenAI. 2025. OpenAI o3 and o4-mini System Card. Technical report, OpenAI. Accessed: 2025-10-06.

Anna Papafragou and Julien Musolino. 2003. Scalar implicatures: experiments at the semantics–pragmatics interface. *Cognition*, 86(3):253–282.

Alicia Parrish, Sebastian Schuster, Alex Warstadt, Omar Agha, Soo-Hwan Lee, Zhuoye Zhao, Samuel R. Bowman, and Tal Linzen. 2021. Nope: A corpus of naturally-occurring presuppositions in english. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 349–366.

Nausicaa Pouscoulous, Ira A. Noveck, Guy Politzer, and Anne Bastide. 2007. A developmental investigation of processing costs in implicature production. *Language acquisition*, 14(4):347–375.

Zhuang Qiu, Xufeng Duan, and Zhenguang Garry Cai. 2023. Pragmatic implicature processing in chatgpt. *PsyArXiv Preprints*.

ZZ Ren, Zhihong Shao, Junxiao Song, Huajian Xin, Haocheng Wang, Wanjia Zhao, Liyue Zhang, Zhe Fu, Qihao Zhu, Dejian Yang, and 1 others. 2025. Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *arXiv preprint arXiv:2504.21801*.

Eszter Ronai and Ming Xiang. 2022. Three factors in explaining scalar diversity. In *Proceedings of sinn und bedeutung*, volume 22, pages 716–733.

Eszter Ronai and Ming Xiang. 2023. Tracking the activation of scalar alternatives with semantic priming. In *Experiments in Linguistic Meaning*, volume 2, pages 229–240.

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. In *Proceedings of the 37th International Conference on Neural Information Processing System*, volume 36, pages 20827–20905.

Sebastian Schuster, Yuxing Chen, and Judith Degen. 2020. Harnessing the linguistic signal to predict scalar inferences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5387–5403.

Dimitrios Skordos and Anna Papafragou. 2016. Children's derivation of scalar implicatures: Alternatives and relevance. *Cognition*, 153:6–18.

Aina Garí Soler and Marianna Apidianaki. 2020. Bert knows punta cana is not just beautiful, it's gorgeous: Ranking scalar adjectives with contextualised representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7371—-7385.

Settaluri Sravanthi, Meet Doshi, Pavan Tankala, Rudra Murthy, Raj Dabre, and Pushpak Bhattacharyya. 2024. Pub: A pragmatics understanding benchmark for assessing llms' pragmatics capabilities. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12075–12097.

John R. Starr, Ashlyn Winship, and Marten van Schijndel. 2025. Generating representations in space with GRIS. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 47, pages 1584–1590.

Bob van Tiel, Emiel Van Miltenburg, Natalia Zevakhina, and Bart Geurts. 2016. Scalar diversity. *Journal of semantics*, 33(1):137–175.

Bob van Tiel and Elizabeth Pankratz. 2021. Adjectival polarity and the processing of scalar inferences. *Glossa: a journal of general linguistics*, 6(1):32.

Bob van Tiel, Elizabeth Pankratz, and Chao Sun. 2019. Scales and scalarity: Processing scalar inferences. *Journal of Memory and Language*, 105:93–107.

Bob van Tiel and Walter Schaeken. 2017. Processing conversational implicatures: alternatives and counterfactual reasoning. *Cognitive science*, 41:1119–1154.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

A Model License and Parameters

Models	License	Parameters	Context
ChatGPT 4o	OpenAI's Terms of Use for Free Users	Unknown	128k tokens
ChatGPT o4-mini-high	OpenAI's Terms of Use for Free Users	Unknown	200k tokens
Gemini 2.5 Pro	Gemini API Terms of Service	Unknown	1M tokens
Gemini 2.0 Flash	Gemini API Terms of Service	Unknown	1M tokens
Claude 3.7	Anthropic's Commercial Terms of Service	Unknown	200k tokens
Qwen2.5 Max	Alibaba Commercial Terms of Usage	Unknown	32k tokens
Qwen3-235B	Apache 2.0	235B	131k tokens
Qwen3-30B	Apache 2.0	31.1B	131k tokens
deepseekR1	MIT	685B	128k tokens
deepseek-prover-v2	deepseek-license	685B	unknown

Table 4: Model licenses and parameters.

B How LLMs generate and judge scales

Pure-Generation1.7k tokens	Rank the words on a scale within each list. [good, bad, excellent, terrible]...(109 items) ...(109 items)
NC-SG 1.7k tokens	The first word and the second word should be positioned next to each other in the ranking. Rank the words on a scale within each list. [good, bad, excellent, terrible] ...(109 items in default order)
NC-RP-POS 3.5k tokens	Rank the words on a scale within each list. [good, bad, excellent, terrible] should be ranked with first word and the third word next to each other. ...(109 items)
NC-RP-Word 3.7k tokens	Rank the words on a scale within each list. [good, bad, excellent, terrible] should be ranked with 'good' and 'excellent' next to each other. ...(109 items)
AC-SG 1.7k tokens	The first word and the third word are antonyms. Rank the words on a scale within each list. [good, bad, excellent, terrible] ...(109 items in default order)
AC-RP-POS 3.5k tokens	Rank the words on a scale within each list. [good, bad, excellent, terrible] should be ranked given that the third word and the fourth word are antonyms. ...(109 items)
AC-RP-Word 3.7k tokens	Rank the words on a scale within each list. [good, bad, excellent, terrible] should be ranked given that 'excellent' and 'terrible' are antonyms. ...(109 items)

Table 5: Prompts for generation tasks. Three variables were manipulated to examine how SOTA LLMs handle constrained scalar reasoning: (1) constraint type (NC/AC), (2) constraint frequency (SG/RP), and (3) referring expression type for the target words (POS/Word). SG = singular constraint occurrence; RP = repeated constraint occurrence; POS = position-based word references. Token numbers indicate average one-shot prompt length.

Judgment	Are the following lists valid orderings to reflect the appropriate relations in each scale? [good, bad, excellent, terrible]...(109 items) ...(109 items) [0.5k tokens]

Table 6: Prompts for judgment tasks.

Models	Pure Generation	Generation with NCs	Generation with ACs
ChatGPT 4o	0.076453	0.179409	0.182467
ChatGPT o4-mini-high	0.944954	0.876656	0.740061
Gemini 2.5 Pro	0.948012	0.958206	0.806575
Gemini 2.0 Flash	0.828746	0.188583	0.497961
Claude 3.7	0.788991	0.636086	0.702345
Qwen2.5 Max	0.694190	0.593612	0.546891
Qwen3-235B	0.155963	0.354400	0.412063
Qwen3-30B	0.657492	0.288141	0.321865
deepseekR1	0.853211	0.545702	0.750255
deepseek-prover-v2	0.886850	0.282025	0.468400

Table 7: Overall model performance accuracy scores in all generation tasks. NC refers to Extra Task Information, and AC refers to Extra Semantic Information.

ChatGPT 4o	N/A	ChatGPT o4-mini-high	**0.913**
Gemini 2.5 Pro	0.7854	Gemini 2.0 Flash	N/A
Claude 3.7	0.7735	Qwen2.5 Max	0.8233
Qwen3-235B	**0.6504**	Qwen3-30B	0.7808
deepseekR1	0.7621	deepseek-prover-v2	N/A

Table 8: Average Accuracy of Ranking Judgment Task (Maximum Accuracy = 1)

C Frequency Correlations Across Tasks

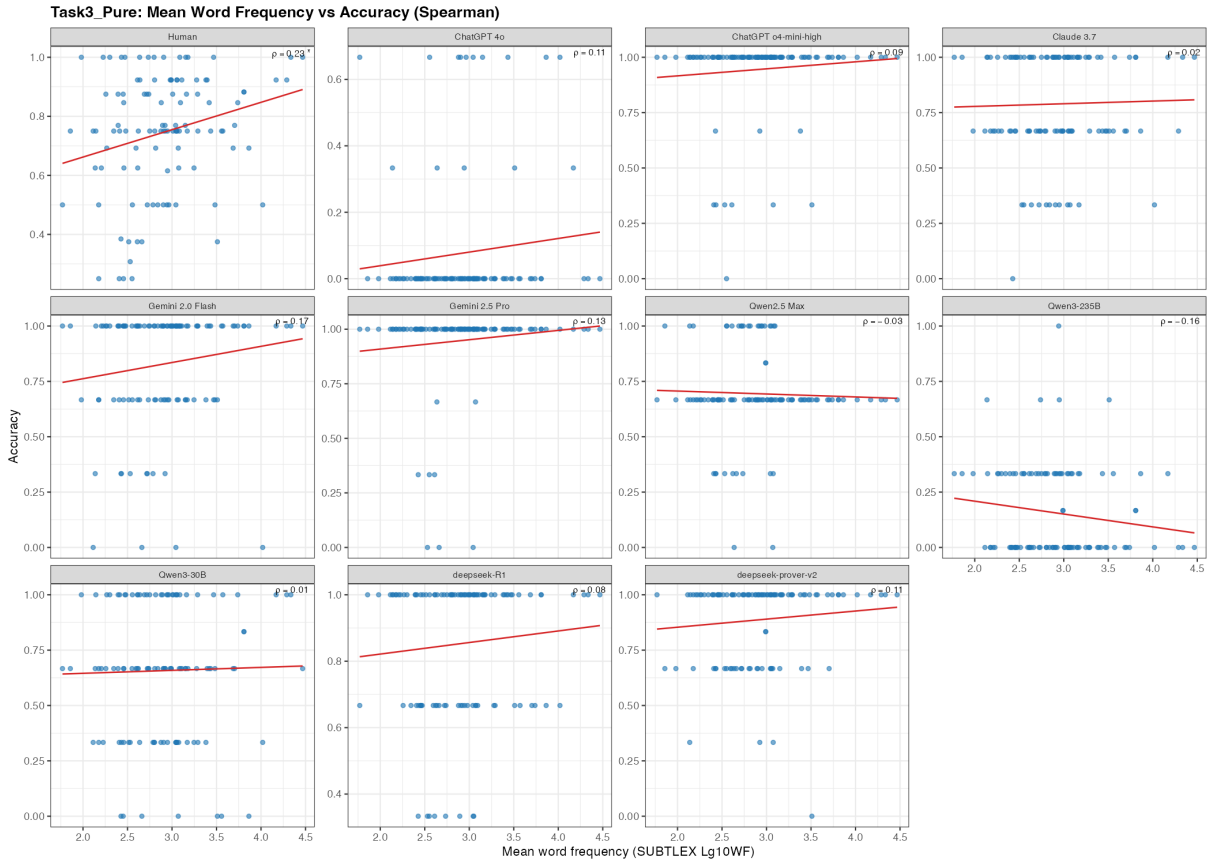


Figure 6: Just-Ranking Experiment: Correlation between Average Word Frequency and Ranking Accuracy of Items

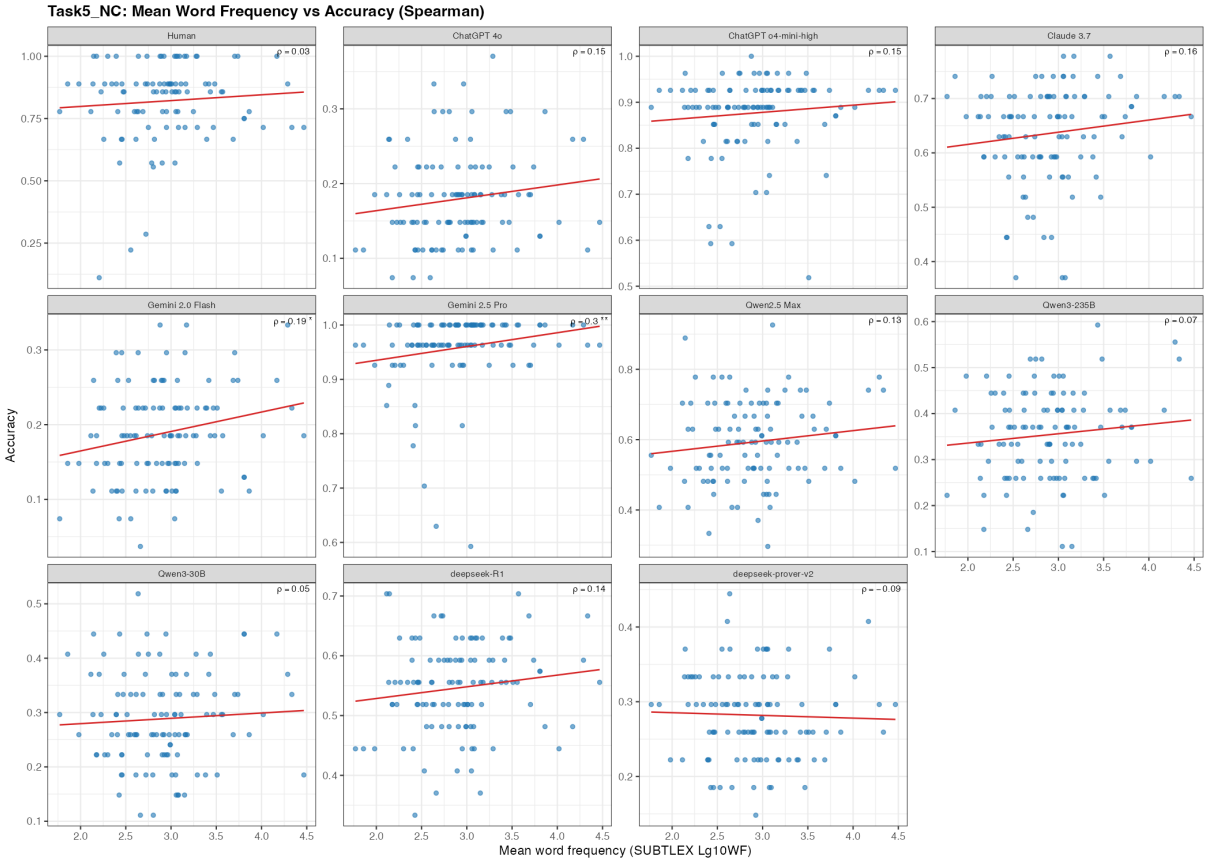


Figure 7: Ranking with Extra Task Information Experiment: Correlation between Average Word Frequency and Ranking Accuracy of Items. NC refers to Extra Task Information.

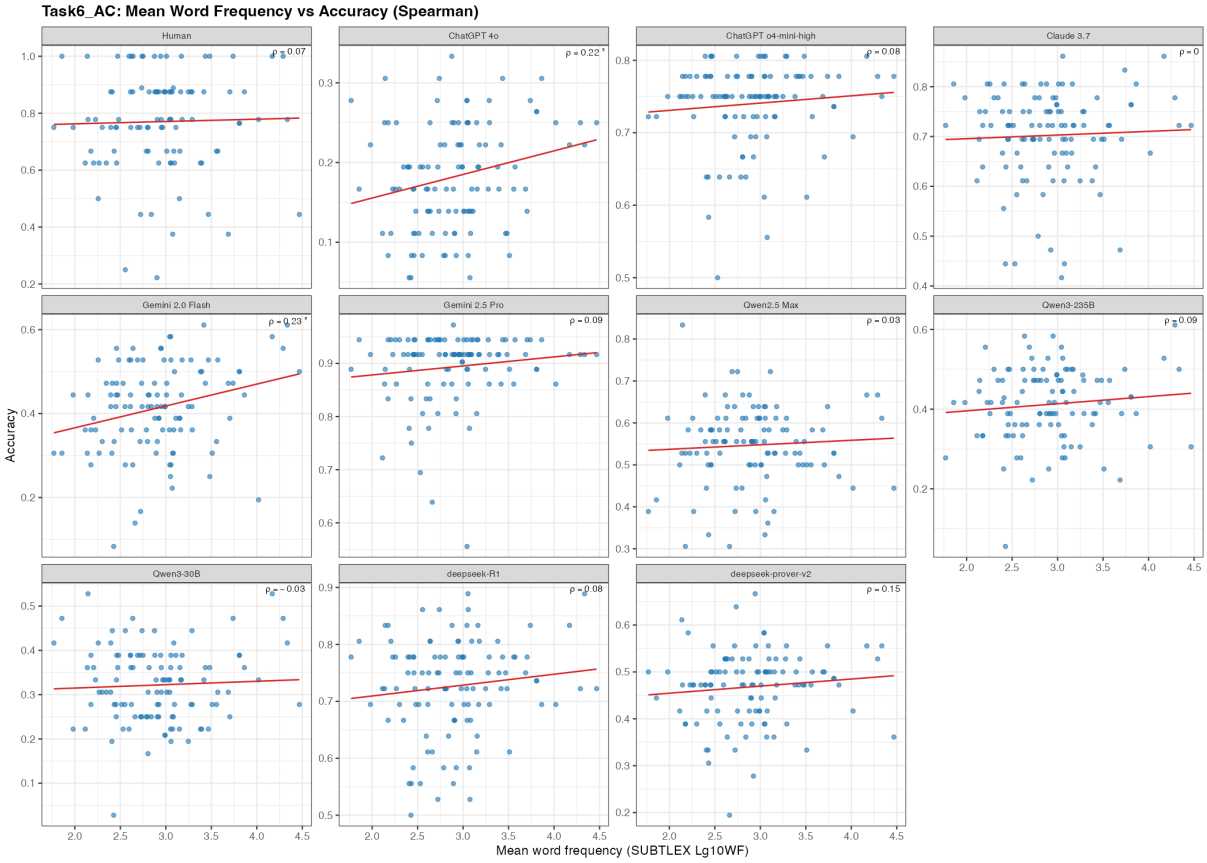


Figure 8: Ranking with Extra Semantic Information Experiment: Correlation between Average Word Frequency and Ranking Accuracy of Items. AC refers to Extra Semantic Information experiment.

D Correlation Analysis

Model	SC	SC_N	SC_A
Qwen3-235B	0.29	0.74^{***}	0.72^{***}
Qwen3-30B	0.5^{*}	0.67^{***}	0.44^{*}
Qwen2.5 Max	0.37	0.73^{***}	0.52^{*}
ChatGPT 4o	0.61^{**}	0.74^{***}	0.77^{***}
o4-mini-high	0.71^{**}	0.76^{***}	0.78^{***}
Gemini 2.0 Flash	0.41	0.64^{**}	0.72^{***}
Gemini 2.5 Pro	0.72^{***}	0.34	0.71^{**}
Claude 3.7	0.59^{**}	0.76^{***}	0.68^{**}
Deepseek-R1	0.57^{*}	0.57^{*}	0.59^{**}
DS-Prover-v2	0.7^{***}	0.7^{***}	0.59^{**}

Table 9: Correlation with Human Invalid Patterns for All Experiments: SC denotes each model's response pattern's spearman correlation to the human baseline. The * symbol denotes p-value thresholds where '***' is $p < 0.001$, '**'< 0.01, and '*'< 0.05. '$_N$' means the metric values are for Extra Task Information experiment, and '$_A$' refers to the Extra Semantic Information experiment. 'o4-mini-high' stands for ChatGPT o4-mini-high and 'DS-Prover-v2' for 'Deepseek-Prover-v2'.

E Barplots for Models Generation Answer types

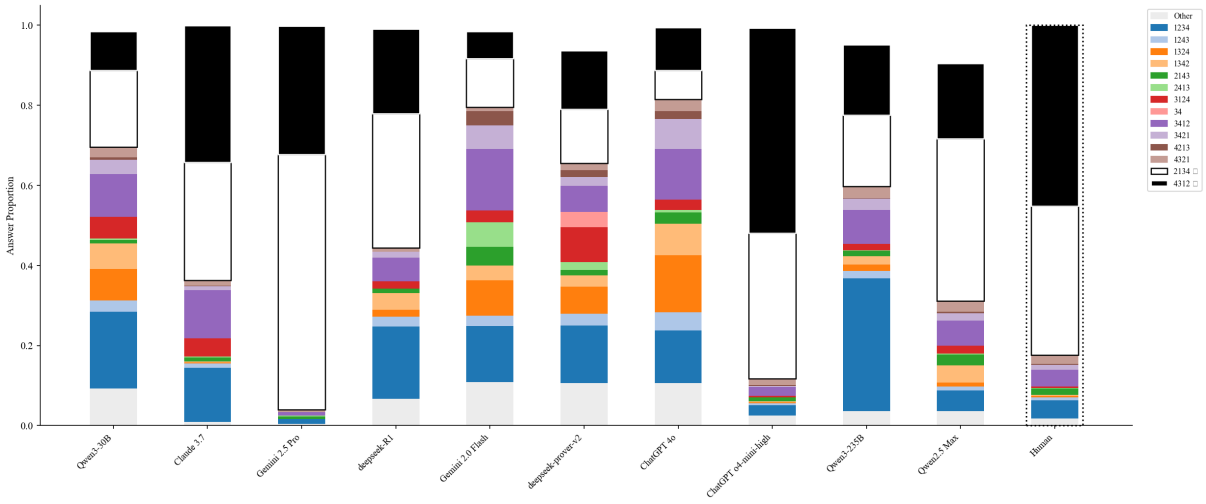


Figure 9: Ranking with Extra Task Info - Response Patterns of LLMs and Humans. Each color bar corresponds to a ranking type.

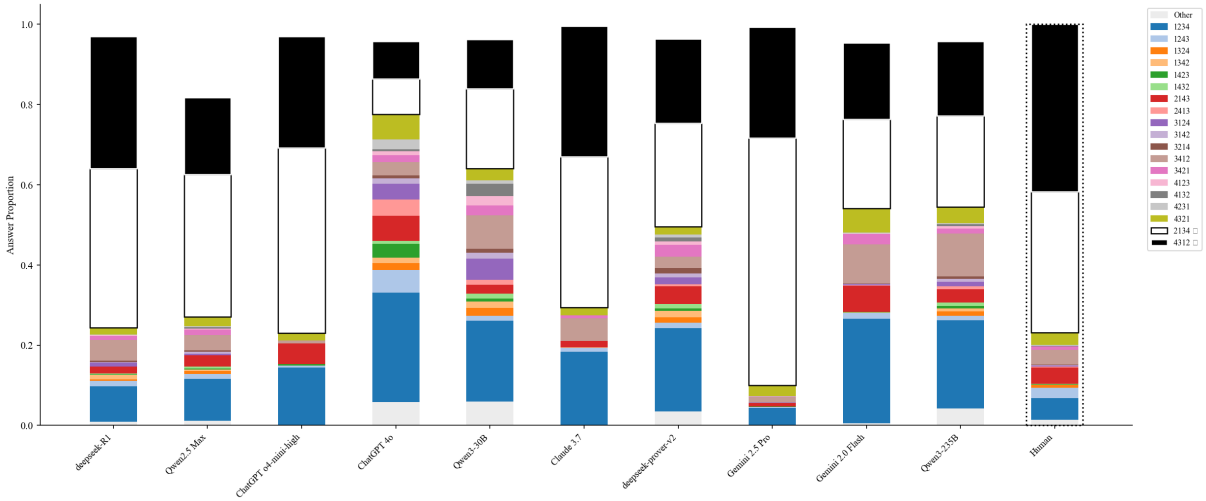


Figure 10: Ranking with Extra Semantic Info - Response Patterns of LLMs and Humans. Each color bar corresponds to a ranking type.

F LLMs Judgment Accuracy

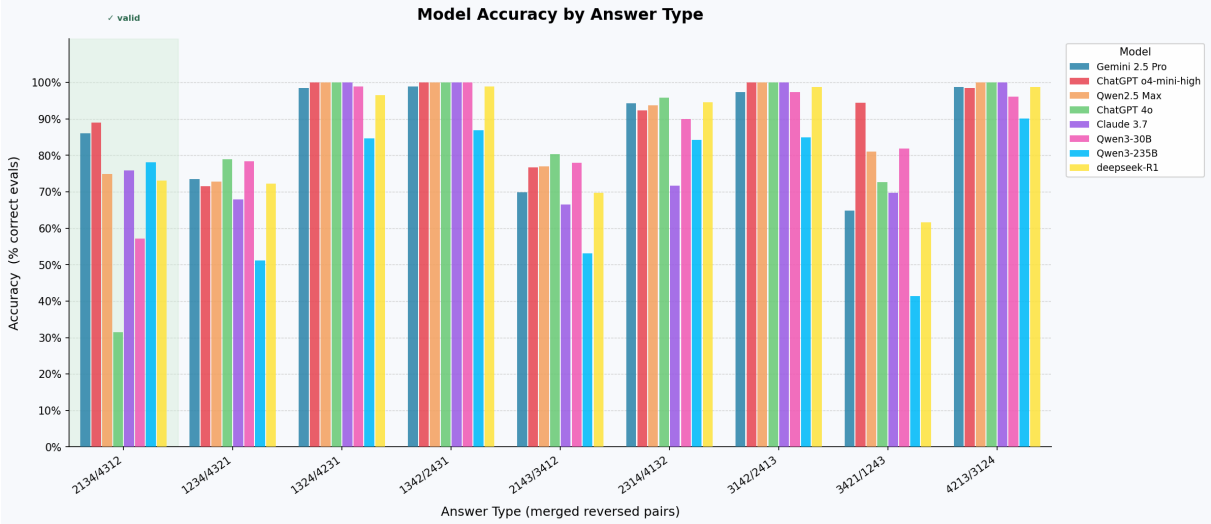


Figure 11: Ten LLMs Judgment Accuracies by Answer Types

G Frequency x Extra Information Interaction

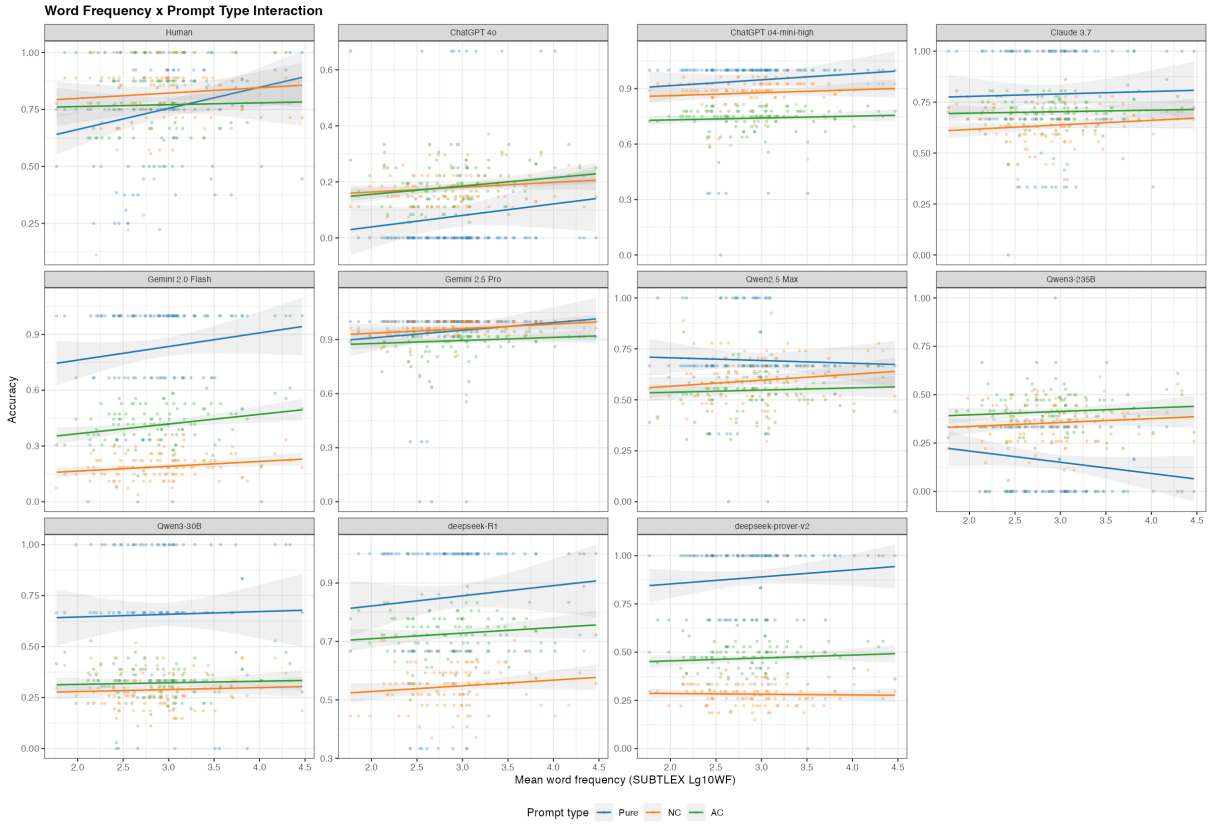


Figure 12: Ten LLMs and human interaction with frequency and extra information. Pure refers to Just-Ranking experiment; NC refers to Extra Task Information experiment; and AC refers to Extra Semantic Information experiment.

H Extra Task vs. Semantic Info

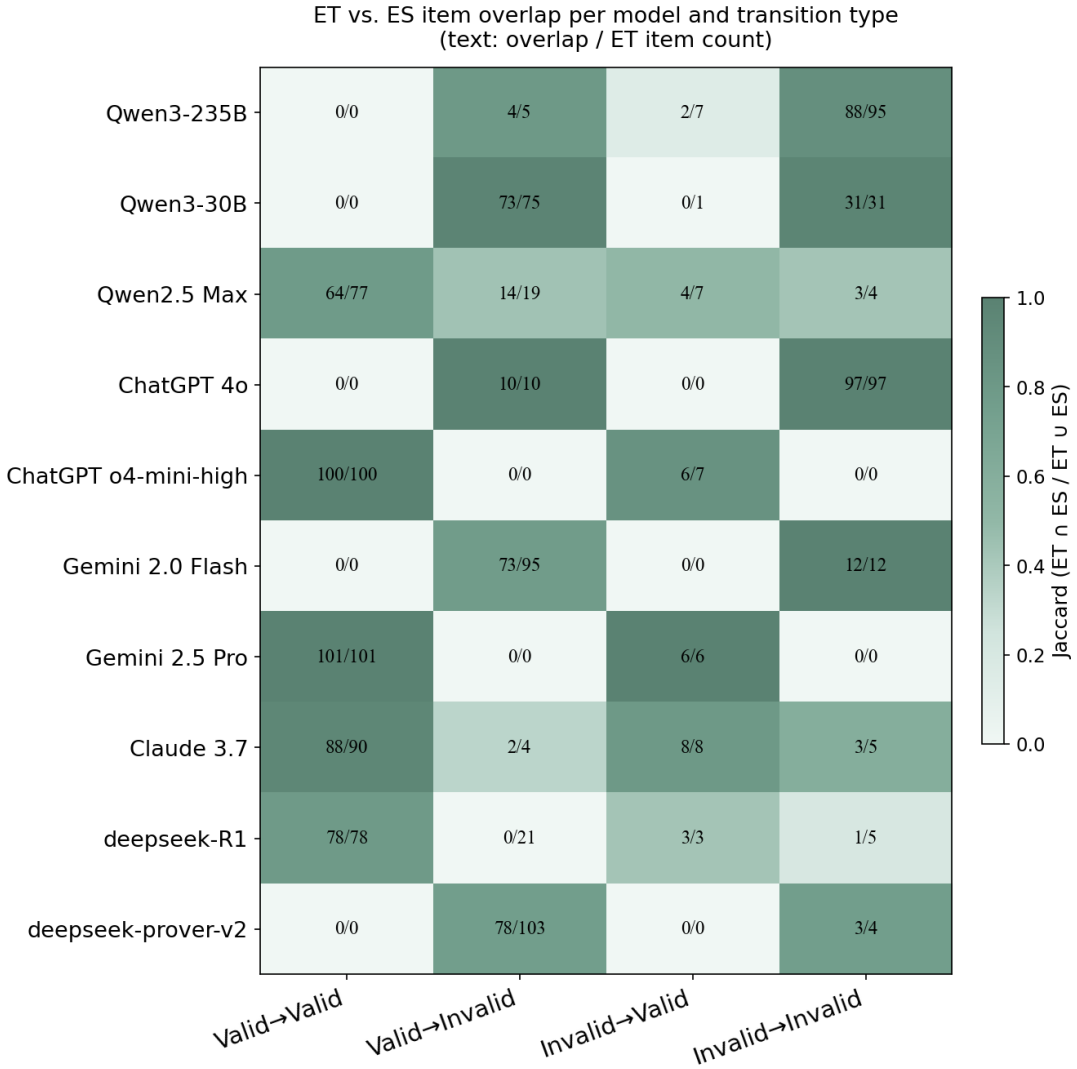


Figure 13: Heatmap of Jaccard Similarity per transition type and model between Extra Task (ET) and Extra Semantic (ES)'s comparison to the Just-Ranking baseline. Darker green means high similarity. The x/y ratio in each box means the number of items shared divided by the union of items in the transition category from both ET/ES transitions from the baseline.

I Within-Language-Model-Family Item Overlap

I.1 Extra Task Info (ET)

Family	Comparison	V→V	V→I	I→V	I→I
DeepSeek	R1 vs. prover-v2	0.00 (0/107)	0.19 (20/107)	0.00 (0/107)	0.00 (0/107)
Gemini	2.0 Flash vs. 2.5 Pro	0.00 (0/107)	0.00 (0/107)	0.00 (0/107)	0.00 (0/107)
ChatGPT	4o vs. o4-mini-high	0.00 (0/107)	0.00 (0/107)	0.00 (0/107)	0.00 (0/107)
Qwen	Q3-235B vs. Q3-30B	0.00 (0/107)	0.05 (4/107)	0.00 (0/107)	0.31 (30/107)
	Q3-235B vs. Q2.5 Mx	0.00 (0/107)	0.09 (2/107)	0.00 (0/107)	0.04 (4/107)
	Q3-30B vs. Q2.5 Mx	0.00 (0/107)	0.13 (11/107)	0.14 (1/107)	0.13 (4/107)

Table 10: Within-family item overlap for the **ET** condition (Just Ranking vs. Extra Task Info). Each cell shows Jaccard (overlap/107) where 107 is the total number of conceptual items per model. Rows within each family are all pairwise model comparisons. 'V' stands fro 'Valid' and 'I' stands for 'Invalid'.

I.2 Extra Semantic Info (ES)

Table 11: Within-family item overlap for the **ES** condition (Just Ranking vs. Extra Semantic Info). Each cell shows Jaccard (overlap/107) where 107 is the total number of list items per model. Rows within each family are all pairwise model comparisons.

Family	Comparison	V→V	V→I	I→V	I→I
DeepSeek	R1 vs. prover-v2	0.23 (23/107)	0.00 (0/107)	0.00 (0/107)	0.00 (0/107)
Gemini	2.0 Flash vs. 2.5 Pro	0.22 (22/107)	0.00 (0/107)	0.00 (0/107)	0.00 (0/107)
ChatGPT	4o vs. o4-mini-high	0.00 (0/107)	0.00 (0/107)	0.00 (0/107)	0.01 (1/107)
Qwen	Q3-235B vs. Q3-30B	0.00 (0/107)	0.04 (3/107)	0.00 (0/107)	0.30 (29/107)
	Q3-235B vs. Q2.5 Mx	0.01 (1/107)	0.07 (2/107)	0.08 (1/107)	0.07 (6/107)
	Q3-30B vs. Q2.5 Mx	0.03 (2/107)	0.19 (16/107)	0.00 (0/107)	0.15 (5/107)

Table 12: Within-family item overlap for the **ES** condition (Just Ranking vs. Extra Semantic Info). Each cell shows Jaccard (overlap/107) where 107 is the total number of list items per model. Rows within each family are all pairwise model comparisons. 'V' stands fro 'Valid' and 'I' stands for 'Invalid'.

J Baseline to Extra Information Item-level Validity Change Analyses

J.1 Valid→Valid

Model	Cond.	n	Both Maj.%	Changed%	Unchanged%	Ambig.%
Qwen3-235B	ET	30	80.0%	6.7%	73.3%	20.0%
	ES	32	53.1%	3.1%	50.0%	46.9%
Qwen3-30B	ET	17	52.9%	5.9%	47.1%	47.1%
	ES	60	31.7%	0.0%	31.7%	68.3%
Qwen2.5 Max	ET	63	76.2%	15.9%	60.3%	23.8%
	ES	105	47.6%	6.7%	41.0%	52.4%
ChatGPT 4o	ET	4	50.0%	0.0%	50.0%	50.0%
	ES	3	0.0%	0.0%	0.0%	100.0%
ChatGPT o4-mini-high	ET	285	76.8%	2.8%	74.0%	23.2%
	ES	208	74.5%	4.8%	69.7%	25.5%
Gemini 2.0 Flash	ET	6	50.0%	33.3%	16.7%	50.0%
	ES	161	64.0%	31.7%	32.3%	36.0%
Gemini 2.5 Pro	ET	295	73.9%	17.6%	56.3%	26.1%
	ES	297	78.5%	22.9%	55.6%	21.5%
Claude 3.7	ET	203	57.1%	2.5%	54.7%	42.9%
	ES	210	60.5%	3.3%	57.1%	39.5%
deepseek-R1	ET	81	32.1%	2.5%	29.6%	67.9%
	ES	219	79.0%	12.3%	66.7%	21.0%
deepseek-prover-v2	ET	33	0.0%	0.0%	0.0%	100.0%
	ES	157	44.6%	0.0%	44.6%	55.4%

Table 13: Valid→Valid analysis (toggle=DIFF: treat reversed rankings as **distinct**). For each model and comparison condition, n is the number of presentations classified as V→V; *Both Maj.%* is the percentage with a clear majority output ranking in *both* Just Ranking and perturbed runs; *Changed%* / *Unchanged%* are subsets of Both Maj.; *Ambig.%* is the remainder. ET = Just Ranking vs. Extra Task Info; ES = Just Ranking vs. Extra Semantic Info.

J.2 Invalid→Invalid

Model	ET: $J_{\text{same}}/J_{\text{diff}}$ $[n]$	ES: $J_{\text{same}}/J_{\text{diff}}$ $[n]$
Qwen3-235B	0.071/0.062 [175]	0.049/0.032 [156]
Qwen3-30B	0.148/0.111 [106]	0.159/0.115 [96]
Qwen2.5 Max	0.041/0.036 [64]	0.048/0.036 [68]
ChatGPT 4o	0.154/0.129 [287]	0.122/0.041 [287]
ChatGPT o4-mini-high	0.358/0.293 [8]	0.324/0.202 [9]
Gemini 2.0 Flash	0.121/0.074 [53]	0.183/0.108 [38]
Gemini 2.5 Pro	0.467/0.233 [5]	0.500/0.333 [2]
Claude 3.7	0.304/0.225 [20]	0.365/0.261 [21]
deepseek-R1	0.175/0.095 [39]	0.246/0.158 [18]
deepseek-prover-v2	0.141/0.086 [37]	0.192/0.159 [24]

Table 14: Invalid→Invalid ranking overlap between conditions. Each cell shows $J_{\text{same}}/J_{\text{diff}}$ $[n]$ where J_{same} is the mean Jaccard treating reversed rankings as identical, J_{diff} treats them as distinct, and n is the number of I→I presentations. Low Jaccard indicates the model gives different wrong rankings across conditions; high Jaccard indicates consistently wrong rankings. ET = Just Ranking vs. Extra Task Info; ES = Just Ranking vs. Extra Semantic Info.

J.3 Valid→Invalid

Model	ET (same)	ET (diff)	ES (same)	ES (diff)
Qwen3-235B	5/17 (29.4%)	2/17 (11.8%)	2/15 (13.3%)	0/15 (0.0%)
Qwen3-30B	11/188 (5.9%)	10/188 (5.3%)	17/145 (11.7%)	4/145 (2.8%)
Qwen2.5 Max	54/153 (35.3%)	40/153 (26.1%)	51/111 (45.9%)	13/111 (11.7%)
ChatGPT 4o	3/21 (14.3%)	3/21 (14.3%)	6/22 (27.3%)	3/22 (13.6%)
ChatGPT o4-mini-high	4/12 (33.3%)	2/12 (16.7%)	48/89 (53.9%)	22/89 (24.7%)
Gemini 2.0 Flash	4/254 (1.6%)	3/254 (1.2%)	42/99 (42.4%)	7/99 (7.1%)
Gemini 2.5 Pro	1/4 (25.0%)	0/4 (0.0%)	1/2 (50.0%)	0/2 (0.0%)
Claude 3.7	3/43 (7.0%)	1/43 (2.3%)	4/36 (11.1%)	2/36 (5.6%)
deepseek-R1	19/186 (10.2%)	12/186 (6.5%)	13/48 (27.1%)	3/48 (6.2%)
deepseek-prover-v2	7/245 (2.9%)	3/245 (1.2%)	25/121 (20.7%)	14/121 (11.6%)

Table 15: Valid→Invalid: percentage of presentations where a dominant (majority, $> 50\%$ of runs) invalid output ranking exists in the perturbed condition. Format: number of items judged as invalid with a dominant ranking / total number of items judged as invalid for each toggle setting (toggle=same when reversed ranking is treated as the same with its reversal counterpart; toggle=diff when reversed ranking like 4321 is treated as different from 1234). A high percentage implies the model is systematically misled to one specific wrong ranking. ET = Just Ranking vs. Extra Task Info; ES = Just Ranking vs. Extra Semantic Info.

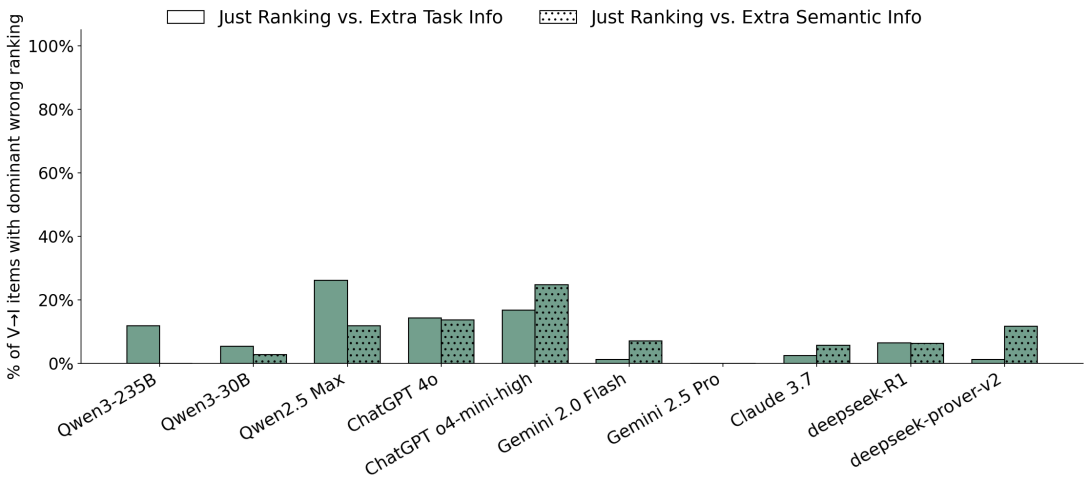


Figure 14: Extra Information Condition Comparison in terms of Valid→Invalid items per model, assuming reversal rankings are treated as different ($1234 \neq 4321$). The higher percentage means the model produced a specific dominant invalid rankings on more items in the extra information condition.

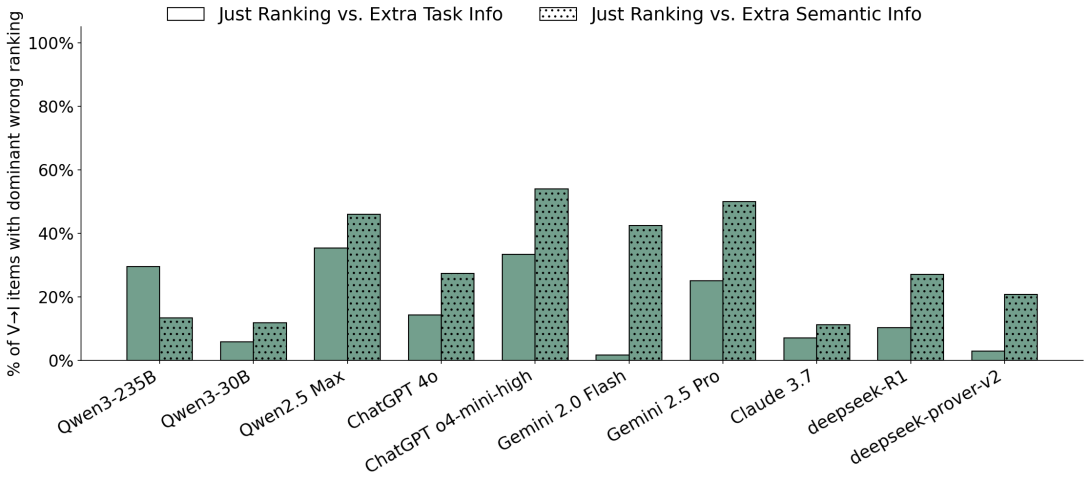


Figure 15: Extra Information Condition Comparison in terms of Valid→Invalid items per model, assuming reversal rankings are treated as the same ($1234 = 4321$). The higher percentage means the model produced a specific dominant invalid rankings on more items in the extra information condition.

J.4 Invalid→Valid

Model	ET (same)	ET (diff)	ES (same)	ES (diff)
Qwen3-235B	38/93 (40.9%)	38/93 (40.9%)	63/112 (56.2%)	63/112 (56.2%)
Qwen3-30B	3/4 (75.0%)	3/4 (75.0%)	12/14 (85.7%)	12/14 (85.7%)
Qwen2.5 Max	6/35 (17.1%)	6/35 (17.1%)	6/31 (19.4%)	6/31 (19.4%)
ChatGPT 4o	3/3 (100.0%)	3/3 (100.0%)	3/3 (100.0%)	3/3 (100.0%)
ChatGPT o4-mini-high	10/10 (100.0%)	10/10 (100.0%)	9/9 (100.0%)	9/9 (100.0%)
Gemini 2.0 Flash	0/2 (0.0%)	0/2 (0.0%)	12/17 (70.6%)	12/17 (70.6%)
Gemini 2.5 Pro	11/11 (100.0%)	11/11 (100.0%)	14/14 (100.0%)	14/14 (100.0%)
Claude 3.7	44/49 (89.8%)	44/49 (89.8%)	42/48 (87.5%)	42/48 (87.5%)
deepseek-R1	8/9 (88.9%)	8/9 (88.9%)	23/30 (76.7%)	23/30 (76.7%)
deepseek-prover-v2	–	–	10/13 (76.9%)	10/13 (76.9%)

Table 16: Invalid→Valid: percentage of presentations where the model had a dominant ($> 50\%$ of Just Ranking runs) invalid output ranking in the *Just Ranking* condition. Format: number of items judged as invalid with a dominant ranking in the baseline / total number of items judged as invalid in the baseline for each toggle setting (toggle=same when reversed ranking is treated as the same with its reversal counterpart; toggle=diff when reversed ranking like 4321 is treated as different from 1234). ET = Just Ranking vs. Extra Task Info; ES = Just Ranking vs. Extra Semantic Info.

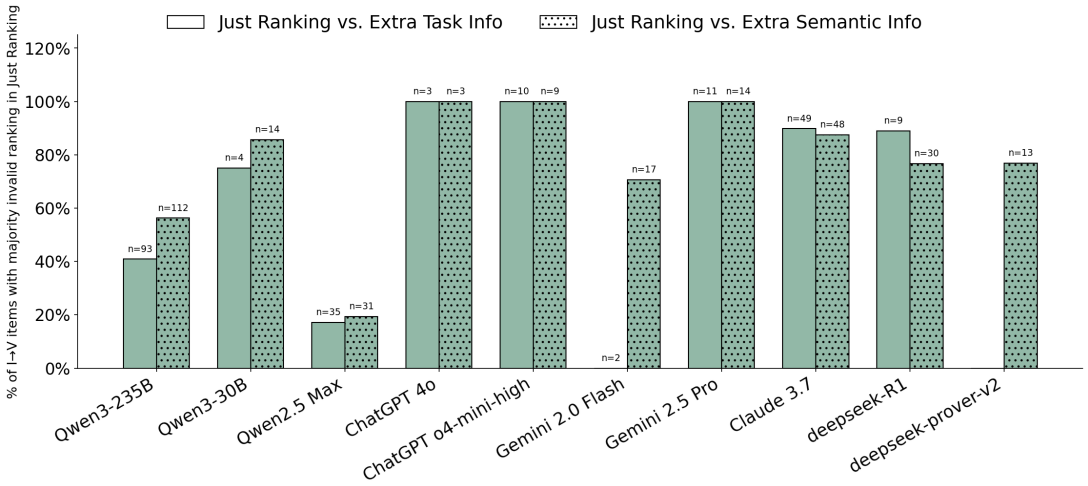


Figure 16: Extra Information Condition Comparison in terms of Invalid→Valid items per model, assuming reversal rankings are treated as different ($1234 \neq 4321$). The higher percentage means the model produced a specific dominant invalid rankings on more items in the baseline condition.

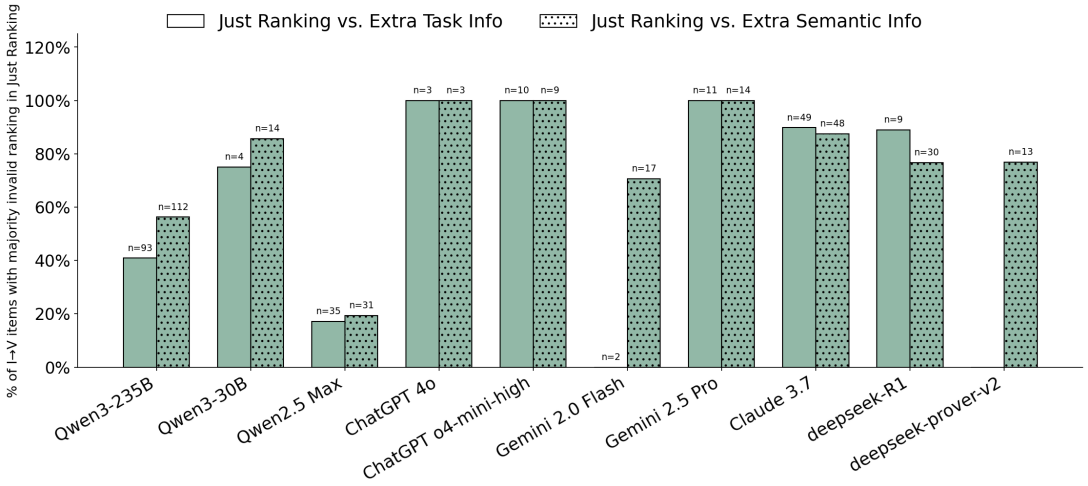


Figure 17: Extra Information Condition Comparison in terms of Invalid→Valid items per model, assuming reversal rankings are treated as the same ($1234 = 4321$). The higher percentage means the model produced a specific dominant invalid rankings on more items in the baseline condition.

Fine-tuning Whisper Across 81 Languages

Shivam Singh
University of California, San Diego
shs046@ucsd.edu

Alex Warstadt
University of California, San Diego
awarstadt@ucsd.edu

1 Introduction

Whisper (Radford et al., 2023) is a family of multilingual automatic speech recognition (ASR) models trained on 680,000 hours of web audio spanning 99 languages. Out of the box, it performs well on high-resource languages like English, Spanish, and French, but its accuracy drops sharply on lower-resource languages. For example, word error rates (WER) exceed 50% for languages like Bengali, Tamil, and Georgian, and reach 100% for Khmer, Burmese, and Lao.

Prior work showed that fine-tuning Whisper on monolingual ASR data can reduce WER by as much as 70% for some languages (Liu et al., 2024), but this approach has yet to be scaled beyond a handful of languages. We fill this gap by fine-tuning Whisper large-v3 on each of the 81 languages in the FLEURS benchmark (Conneau et al., 2023). We find that fine-tuning improves speech recognition for all 81 languages tested, reducing WER by nearly 30% on average. However, the size of the improvement varies widely, and we show that the language's writing system is the best predictor of how much fine-tuning helps. We further demonstrate that this script-family pattern is mediated by tokenizer fertility: Whisper's BPE compression ratio (characters per token) correlates with WER reduction at Spearman $\rho \approx -0.78$, and the dependence persists after partialing out per-language pretraining hours. We intend to release model weights upon publication, providing, to the best of our knowledge, the largest fleet of open-source language-specialized ASR models.

2 Methods

We fine-tune Whisper large-v3 independently for each of the 81 FLEURS languages. Every language uses the same pipeline: load a fresh copy of the pretrained model, train on the FLEURS training split, evaluate on the validation split, and save the checkpoint with the lowest WER. Early stopping triggers after 3 epochs without improvement or 12 epochs total. Audio is resampled to 16 kHz, capped at 30 s, and trained with AdamW, gradient clipping at 1.0, and mixed precision on a single A100. We use separate learning rates for the encoder and decoder, setting the encoder learning rate to $0.3\times$ the decoder learning rate (this ratio was consistent across both hyperparameter sweeps).

Hyperparameter	Config A	Config B
Learning rate	1.18×10^{-5}	6.48×10^{-6}
Grad. accumulation	24	16
LR scheduler	Exponential	Cosine
LR decay rate	0.9	0.95
Encoder LR ratio	0.3	
Warmup steps	150	
Max epochs	12	
Early stopping	Patience 3	

Table 1: Hyperparameter configurations. Values below the divider are shared across configs.

Two hyperparameter configurations were used (Table 1), found by Bayesian sweeps on Hindi and German. These two languages were chosen as endpoints of Whisper's coverage spectrum—Hindi (Devanagari, mid-coverage) and German (Latin, high-coverage)—to test whether a single set of hyperparameters generalizes across script families. The two configurations correlate at $\rho = 0.996$ on the resulting 81-language WER vector, indicating that our conclusions are robust to hyperparameter choice within a reasonable range.

3 Results

Figure 1 plots zero-shot WER from pretrained Whisper large-v3 against fine-tuned WER for each target language under both configurations. Every point falls below the diagonal, meaning fine-tuning helps across the board. Under Config A, the mean WER drops from 34.8% to 24.6%, and 50 out of 81

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 408–410
July 3-4, 2026 ©2026 Association for Computational Linguistics

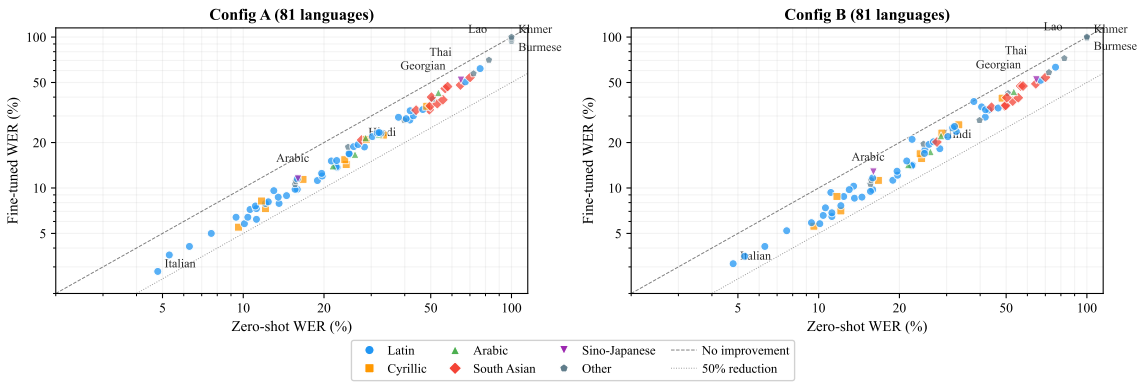


Figure 1: Zero-shot vs. fine-tuned WER under Config A (left) and Config B (right). Points are colored by script family. The dashed line marks no improvement; the dotted line marks 50% WER reduction. The two configurations produce nearly identical patterns (Spearman $\rho = 0.996$).

languages reach a fine-tuned WER below 25%.

The writing system is a strong predictor of improvement. Latin and Cyrillic scripts see 30–45% WER reductions, reaching single-digit error rates for Italian (2.8%), Spanish (3.6%), and Russian (5.5%). Brahmic scripts (South Asian languages) improve ≈25% relative but remain at 20–54% WER. Languages with unique or logographic scripts benefit least: Thai drops only 14% (82.4%→70.5%), Burmese 3%, and Khmer not at all.

Script	N	Mean FT	Rel. ↓	FT Range
Latin	44	16.2%	32.8%	2.8–61.8%
Cyrillic	9	15.6%	34.3%	5.5–34.8%
Arabic	5	23.5%	29.0%	14.0–42.8%
Brahmic	11	39.2%	25.7%	20.8–54.0%
Sino-Japanese	2	31.7%	24.0%	11.5–51.9%
Other	10	52.7%	18.1%	10.6–100.1%

Table 2: Fine-tuned WER by script family (Config A). Rel. ↓ = mean relative WER reduction from zero-shot.

Table 2 summarizes these trends by script family. This pattern holds even for poorly represented language families: Vietnamese (Austroasiatic, Latin script) reaches 7.6% WER, while Thai (Kra–Dai, unique script) stays at 70.5%. Similarly, Lingala (Niger-Congo, Latin) reaches 15.2%, while Georgian (Kartvelian, unique script) only reaches 57.3%. Config B produces nearly identical results: mean WER 25.6% vs. 24.6% for Config A, Spearman $\rho \approx 0.996$ ($p < 10^{-74}$) between the two, and a mean absolute difference of just 1.1 pp.

Tokenizer compression ratio predicts fine-tuning headroom. We directly link the script-family pattern to tokenizer behavior. For each language, we compute Whisper's BPE compression ratio (Unicode characters per token) on the FLEURS test transcripts. Across our 81 languages, this compression ratio correlates with relative WER reduction at Spearman $\rho \approx -0.78$: scripts that Whisper's tokenizer splits into many short fragments (low chars/token) gain less from fine-tuning. The same finding holds when controlling for Whisper's per-language pretraining duration via partial Spearman correlation (partial $\rho \approx -0.79$), indicating that the effect is not simply "less pretraining data → less benefit" but specifically reflects source-tokenizer fit.

4 Discussion

Our results replicate and scale up Liu et al. (2024), confirming that monolingual fine-tuning consistently improves Whisper across all 81 FLEURS languages, and suggest a simple solution to the curse of multilinguality (Conneau et al., 2020): recover strong monolingual performance while leveraging multilingual pretraining via fine-tuning.

The writing-system predictor, together with the compression ratio correlation, points to tokenization as Whisper's main bottleneck. Whisper's BPE vocabulary is trained predominantly on English text (Radford et al., 2023), so Latin and Cyrillic languages share many subword tokens with the existing vocabulary, and fine-tuning need only adjust acoustic-to-token mappings. In contrast, a Bengali sentence requiring 15 tokens with a dedicated tokenizer may require 40+ with Whisper's vocabulary, making learning harder and inference slower. For the roughly 20–30 languages where fine-tuned

WER remains above 30%, tokenizer adaptation or script-aware pretraining may be needed.

Future work. In ongoing work, we are extending this analysis to evaluate vocabulary-replacement and related tokenizer-adaptation methods as a direct remedy for poorly tokenized scripts (manuscript in preparation). Preliminary results suggest that swapping Whisper's tokenizer for a language-specific BPE recovers substantial headroom on the highest-fertility scripts, including languages that vanilla fine-tuning leaves above 50% WER.

References

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.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riber, Ankur Bapna, and 1 others. 2023. FLEURS: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805. IEEE.

Yunpeng Liu, Xukui Yang, and Dan Qu. 2024. Exploration of Whisper fine-tuning strategies for low-resource ASR. *EURASIP Journal on Audio, Speech, and Music Processing*, 2024(1):29.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR.

A Transparent Model of Syntactic and Semantic Cue-based Retrieval

Shisen Yue
Johns Hopkins University
syue11@jh.edu

John Hale
Johns Hopkins University
jthale@jhu.edu

Abstract

Human comprehenders have greater difficulty forming pairwise grammatical dependencies in cases where the earlier word competes with a "distractor" to which it is similar. Cue-based retrieval theories (see e.g., Lewis et al., 2006) address this "interference" phenomenon with explicit quantifications of memory retrieval difficulty. We propose a computational model, consistent with Cue-based retrieval, that separately quantifies two different kinds of similarity. A linear combination of the two reproduces the graded interference pattern reported in Van Dyke (2007). This simple account offers a more straightforward mechanistic interpretation than Attention-based predictors from opaque Transformer based models.

1 Introduction

A longstanding question in human language processing is how comprehenders retrieve previously encountered words from memory to establish dependencies with the current word. Cue-based retrieval theory proposes that memory access is content-addressable: morphological, syntactic, and semantic cues from the current word probe memory for matching candidates, while partially matching distractors create similarity-based interference during retrieval (for an overview, see Lewis et al., 2006).

Recent work has drawn an analogy between this retrieval process and the attention mechanism in Transformer language models. Under this view, a word's features act as queries that retrieve matching keys from the preceding context, and attention patterns provide candidate predictors of retrieval success and competition (Ryu and Lewis, 2021; Oh and Schuler, 2022; Ryu and Lewis, 2025). While such predictors often correlate with human processing difficulty, their interpretability remains limited: attention signals are distributed across many layers and heads, and there is no foolproof way to identify which components correspond to specific linguistic operations (Clark et al., 2019; Voita et al., 2019; Jain and Wallace, 2019).

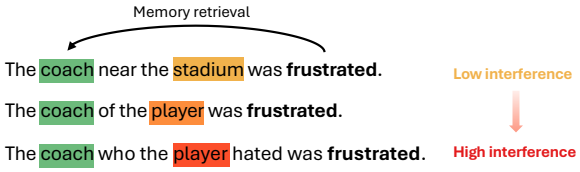


Figure 1: Retrieval of the subject *coach* becomes harder as the distractor increasingly matches the target in relevant linguistic features, producing a gradient from low to high interference. The full model computation for this example is shown in Appendix Figure 9.

One response has been to build smaller models that parameterize retrieval computations more explicitly. For example, Timkey and Linzen (2023) implement query–key–value–style retrieval operations while constraining model capacity to approximate human working-memory limits. However, because retrieval signals are still implemented through distributed neural dynamics, the mapping between model parameters and specific linguistic properties remains difficult to characterize mechanistically.

In this spirit, we ask whether the interaction between syntactic structure and semantic plausibility can be explained using a simple, explicitly parameterized model rather than opaque neural architectures. To investigate this possibility, we develop a lightweight linear model that factors syntactic and semantic contributions to interference while remaining close to the cue-based retrieval theory. Our approach treats each dependency arc as a retrieval trace and models retrieval as an asymmetric linear transformation from a cue representation into the target's representational space. The syntactic component (Grammar-bilinear model) uses supertags from Combinatory Categorial Grammar (CCG), a lexicalized syntactic theory whose

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 411–422
July 3-4, 2026 ©2026 Association for Computational Linguistics

word-level supertags provide fine-grained syntactic descriptions that has been argued to align well with human sentence comprehension (Steedman, 2001; Steedman and Baldridge, 2011; Stanojević et al., 2023). The semantic component (Asymw2v) adapts dependency-based embeddings (Levy and Goldberg, 2014) to learn distinct input (cue) and output (context-bound target) spaces. By separating these two sources of competition, the model provides a transparent parameterization of interference that can be analyzed directly in terms of syntactic structure and lexical compatibility. We evaluate the model on the interference materials of Van Dyke (2007), which manipulate syntactic similarity and semantic plausibility in a 2×2 design. Our results show that: (1) the syntactic and semantic submodels produce independent interference signals corresponding to structural similarity and thematic plausibility; (2) a simple linear combination of these signals yields graded retrieval predictions across the four experimental conditions; and (3) the resulting mixed model outperforms attention heads within GPT-2 that are putatively specialized for grammatical subjecthood, both in retrieval accuracy and in alignment with human reading difficulty.

2 Background

Cue-based memory retrieval theory Cue-based retrieval theory proposes that sentence comprehension involves retrieving information that was introduced by earlier words using features of the current words as memory cues. Morphological, syntactic, and semantic properties of the cue probe a content-addressable memory store to retrieve the most compatible candidate (Lewis et al., 2006; Parker et al., 2017; Vasishth and Engelmann, 2021). Interference arises when multiple items partially match these cues, leading to slower processing and increased comprehension difficulty. Computational implementations such as Lewis and Vasishth (2005) characterize retrieval latency and accuracy as a function of cue–target similarity and competition. In syntactic dependency processing, each dependency arc can be viewed as a retrieval event: when encountering a verb such as *frustrated*, the processor must retrieve its subject (e.g., *coach*) from preceding context. Interference occurs when distractors (e.g., *player*) share syntactic or semantic features with the intended target, producing measurable effects on reading times and comprehension accuracy (Van Dyke, 2007; Oberauer and Lewandowsky, 2008).

Distributional representations Cue-based retrieval theory distinguishes between lexical information that serves as a memory cue and working-memory elements that are retrieved. In this work, we operationalize both cues and candidate memory elements as word embeddings, making distributional representations a natural way to model lexical retrieval. Distributional embedding models learn vector representations in which words that occur in similar contexts occupy nearby regions of a shared space. Saussure (1916) distinguishes between syntagmatic relations, in which words are associated with the words that surround them, and paradigmatic relations, in which words are related because they can substitute for one another. Building on this distinction, Kelly et al. (2020) show that higher-order distributional associations can induce syntactic lexical categories, such as part-of-speech and CCG syntactic types. Our approach instead focuses on the syntagmatic side of lexical knowledge: retrieval is modeled as an asymmetric relation between a cue and a role-bound target. Dependency-based word embeddings (Levy and Goldberg, 2014) provide a natural implementation of this idea by defining contexts not as linear windows but as syntactic dependency relations extracted from parsed text. Each training instance pairs a word with a relation-labeled context (e.g., *coach/nsubj* represents *coach* when appearing as a candidate word to be retrieved in a `nsubj` relation), allowing the model to capture functional relations (e.g., dog $\leftrightarrow$ bark) between words rather than topical co-occurrence (dog $\leftrightarrow$ paw). As in the Skip-Gram framework (Mikolov et al., 2013), the model learns two parameter matrices: an input matrix representing words when they serve as cues (queries) and an output matrix representing relation-bound contexts.

LLMs as psycholinguistic predictors Recent work has used autoregressive language models to derive predictors of human sentence processing difficulty, typically operationalized as next-word surprisal. Transformer-based models such as GPT-2 reliably capture well-known effects in reading behavior and have been shown to predict reading times and neural responses during language comprehension (Shain, 2024; Caucheteux et al., 2022). However, several studies report an inverse scaling effect: smaller models often correlate better with

human reading-time data than larger ones, possibly because larger models form overly confident expectations for rare or unpredictable words (Arehalli et al., 2022; Oh and Schuler, 2023b,a; Steuer et al., 2023). This observation has motivated approaches that derive psycholinguistic predictors from internal model representations rather than output probabilities alone. For example, attention-based entropy and distance metrics (Oh and Schuler, 2022) and analyses of syntactically specialized attention heads (Ryu and Lewis, 2021, 2025) attempt to link model components more directly to retrieval-like processes in human sentence comprehension. More broadly, recent work emphasizes cognitively constrained or lightweight architectures as potentially better aligned with human incremental processing mechanisms (Clark et al., 2025; Timkey and Linzen, 2023).

3 Model

We model interference during cue-based retrieval in sentence processing. Retrieval is treated as the operation that links the current word (the cue) to a previously encountered word (the target) in memory, corresponding to dependency relations in the sentence (Van Dyke and McElree, 2006; Hofmeister and Vasishth, 2014). We formalize retrieval as an asymmetric transformation that projects a cue representation into the representational space of candidate targets and produces a probability distribution over preceding words, reflecting their relative retrieval scores.

The model separates syntactic and semantic sources of interference using two components. Section 3.1 introduces the grammar-bilinear model, which approximates syntactic compatibility using CCG-based features. Section 3.2 presents the Asymw2v lexical model, which models semantic selectional preferences using dependency-based embeddings. Section 3.3 then combines these signals into a mixed model. While the present study focuses on the `nsubj` dependency as a case study, the framework naturally extends to other dependency relations.

3.1 Grammar-bilinear model

The grammar-bilinear model captures syntactic interference by scoring the compatibility between a cue word and candidate targets in its left context. Retrieval is formulated as a bilinear similarity between cue and candidate representations, producing a probability distribution over preceding words.

the	coach	of	the	player	was	frustrated
NP/N	N	$(NP\backslash NP)/NP$	NP/N	N	$(S[dcl]\backslash NP)/(S[adj]\backslash NP)$	$S[adj]\backslash NP$

Figure 2: Example of CCG supertags assigned to a sentence. Each word is annotated with a lexical category encoding its combinatory behavior. For example, the verb *was* receives the category $(S[dcl]\backslash NP)/(S[adj]\backslash NP)$, indicating that it combines with an adjectival predicate on the right to form a verb phrase that in turn combines with a subject noun phrase on the left.

Representation The model's syntactic information is limited to CCG supertags. These tags encode combinatory potential including subcategorization. Because these categories are highly informative about syntactic composition, assigning supertags is often described as "almost parsing" (Bangalore and Joshi, 1999). An example of supertag assignments is shown in Figure 2, where each word in the sentence is annotated with a lexical category describing how it combines with neighboring constituents. Supertag therefore provide a rich feature space for modeling fine-grained combinatory constraints in machine-learning models (Kasai et al., 2019).

Motivated by these properties, we approximate the syntactic environment of a word k as the concatenation of the supertags of the current word and its two preceding words:

$$h_k = [G_{k-2}; G_{k-1}; G_k]$$

where G_i is a one-hot vector over the supertag vocabulary. Training data are drawn from the English Web Treebank, using CCG supertags from *ccgtools*[1] and dependency relations from the Universal Dependencies annotations (Silveira et al., 2014).

Retrieval scoring Given cue representation h_k, a trainable matrix W projects the cue into the candidate space. If the supertag vocabulary has size V, then $h \in \mathbb{R}^{3V}$ and $W \in \mathbb{R}^{3V \times 3V}$. For each candidate c in the preceding context, the retrieval score is

$$s_c = h_c^\top W h_k$$

Scores are normalized over all preceding words using a softmax:

$$\hat{p}(w_c|w_k) = \frac{e^{h_c^\top W h_k}}{\sum_{i<k} e^{h_i^\top W h_k}}$$

[1] https://github.com/stanojevic/ccgtools

The model is trained to maximize the likelihood of retrieving the correct subject specified by the dependency parse. Because supertags encode detailed combinatory constraints, the model learns to assign higher scores to candidates occupying syntactically compatible positions, while structurally incompatible candidates receive lower retrieval probabilities.

3.2 Asymw2v

The Asymw2v model captures semantic interference through dependency-based distributional representations. It models the compatibility between a cue word and candidate targets using embeddings trained on dependency-labeled contexts, allowing the model to encode selectional preferences associated with particular grammatical relations.

Representation and parameters Following the dependency-based embedding framework introduced in Section 2, each training pair includes a word and a relation-labeled context derived from the dependency arcs (e.g., *frustrated*, *coach/nsubj*). The model learns two sets of embeddings: an input embedding $v_w \in \mathbb{R}^d$ for each word w, and an output embedding $u_c \in \mathbb{R}^d$ for each relation-bound context c. If the word vocabulary has size $|V|$ and the set of dependency contexts has size $|C|$, the model contains $d(|V| + |C|)$ parameters.

Embeddings are trained using the Skip-gram objective on a dependency-parsed Wikipedia corpus[2]. After training, the input embedding of the current word serves as the cue vector, while the output embeddings of relation-labeled candidates (e.g., *coach/nsubj*) serve as role-specific target representations.

Retrieval scoring Given cue embedding v_k, the compatibility between the cue and a candidate context c is computed using a dot product

$$s_c = v_k^\top u_c$$

and normalized across candidate targets using a softmax over all preceding words in the sentence. Because contexts are tied to specific dependency relations, the model learns graded preferences for plausible fillers of each relation. For example, verbs that frequently take animate subjects assign higher scores to animate `nsubj` candidates than to inanimate distractors, providing a distributional signal for semantic interference during retrieval.

[2] https://huggingface.co/datasets/wikimedia/wikipedia

3.3 Mixed model

The mixed model combines syntactic and lexical retrieval signals by linearly combining their scores before normalization. For each candidate i,

$$s_i^{mix} = \alpha s_i^{syn} + \beta s_i^{lex}$$

where s_i^{syn} and s_i^{lex} are the scores produced by the grammar-bilinear and Asymw2v models respectively. The resulting scores are converted into retrieval probabilities using a softmax over all preceding candidates:

$$p_i^{mix} = \text{softmax}(s^{mix})$$

This late-fusion formulation reflects cue-based retrieval accounts in which syntactic constraints provide the primary retrieval cues while semantic compatibility contributes graded competition among candidates (Van Dyke and Lewis, 2003; Van Dyke and McElree, 2011). By combining the two signals at the scoring stage, the model preserves strong syntactic filtering while allowing lexical-semantic similarity to modulate retrieval interference.

4 Experiments

We evaluate whether the proposed mixed model (as well as its two subcomponents) derive the inference patterns that we would expect under cue-based retrieval theory. Specifically, we ask whether the grammar-bilinear component captures syntactic interference, whether Asymw2v captures semantic interference, and whether their combination recovers the full human-like interaction between the two.

4.1 Experimental Materials

We evaluate the model using the sentence materials from Tan et al. (2017), adapted from Van Dyke (2007). These stimuli implement a 2×2 factorial design crossing syntactic interference (Low vs. High) and semantic interference (Low vs. High). Each item set contains four sentences corresponding to the conditions LoSynLoSem, LoSynHiSem, HiSynLoSem, and HiSynHiSem.

In these materials, the retrieval cue occurs at the main verb (e.g., *will forget*), which retrieves its grammatical subject (*the hostess*). Syntactic interference is manipulated by varying whether the intervening noun phrase occupies subject or object position in the embedded clause. Semantic interference is manipulated by varying the plausibility of the distractor noun as the subject of the main verb

(e.g., *toddler* vs. *room*). High interference arises when the distractor overlaps with the retrieval cues in syntactic or semantic features.

Table 1 illustrates the structure of the stimuli.

Sentence Region	Example Stimulus
Introduction	The hostess
Intervening region	
LoSyn/LoSem	who had yelled about the dirty **room** loudly
LoSyn/HiSem	who had yelled about the dirty **toddler** loudly
HiSyn/LoSem	who yelled that the **room** was dirty loudly
HiSyn/HiSem	who yelled that the **toddler** was dirty loudly
Critical region	will forget
Spillover region	about the
Last word	mess

Table 1: Experimental materials from Tan et al. (2017). Interference words are bold and differ in their syntactic/semantic compatibility to the verb.

4.2 Experimental Setup

The mixed model introduced in Section 3.3 combines the syntactic and lexical components with fixed weights $\alpha = 0.4$ and $\beta = 0.6$, which are selected without an extensive hyperparameter search for a reason to provide a simple, fixed combination of the two components.

To quantify retrieval competition, we compute the relative retrieval score $Relret$ of the correct subject. Following Timkey and Linzen (2023), this measure represents the proportion of retrieval probability assigned to the correct target relative to the distractor:

$$RelRet(v, s) = \frac{Ret(v, s)}{Ret(v, s) + Ret(v, d)}$$

where $Ret(v, s)$ denotes the model's retrieval score from cue v to the correct subject s, and d denotes the distractor candidate. Given the balanced 2×2 design, a two-way repeated-measures ANOVA is also performed to assess main effects of syntactic and semantic interference and their interaction.

4.3 Experiment 1: Component Validation

We first test whether the two model components independently capture their intended interference dimensions. The grammar-bilinear model should exhibit sensitivity to syntactic interference, while the Asymw2v model should respond primarily to semantic interference.

Figure 3 shows the relative retrieval scores predicted by the two models across the experimental conditions. As expected, the grammar-bilinear model shows a strong effect of syntactic interference, with significantly lower retrieval scores in HiSyn conditions than LoSyn conditions, remaining insensitive to semantic interference. In contrast, the Asymw2v model shows that retrieval scores decrease under high semantic interference but remain unaffected by syntactic interference. The paired t-test statistics for these contrasts demonstrate that all predicted effects are statistically significant ($p < 0.05$), confirming that the two components capture distinct sources of retrieval interference.

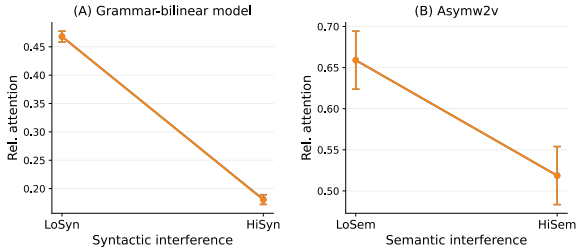


Figure 3: Separate validation of the two components. Mean relative retrieval scores predicted by the grammar-bilinear model and Asymw2v. The grammar-bilinear model (A) is sensitive to syntactic interference, whereas the Asymw2v model (B) is sensitive to semantic interference.

4.4 Experiment 2: Mixed Model Evaluation

Table 2 compares retrieval accuracy for the mixed model, two GPT-2-small attention heads, and human data. A prediction is correct when the highest retrieval or attention score is assigned to the intended subject. The GPT-2 heads are identified by Ryu and Lewis (2021) as `nsubj`-sensitive using the head-specialization diagnostic of Voita et al. (2019).

Our model captures the human-like accuracy hierarchy, with highest accuracy in LoSynLoSem and lowest accuracy in HiSynHiSem. The Transformer heads fail in complementary ways: $head4_3$ robustly attends to the correct subject under low syntactic interference but shows little semantic sensitivity, whereas $head3_6$ produces more graded condition-wise relative-attention scores but often assigns attention to the wrong item, yielding near-zero retrieval accuracy.

Figure 4 shows the same contrast in relative retrieval or attention scores, calculated following Timkey and Linzen (2023). The mixed model shows significant syntactic and semantic interference effects, with all pairwise comparisons signif-

Condition	Mixed model	Attention heads (Ryu and Lewis, 2021)		Human (Van Dyke, 2007)
		head4_3	head3_6	
LoSynLoSem	0.5625	0.4250	0.0250	0.90
LoSynHiSem	0.4500	0.4625	0.0375	0.82
HiSynLoSem	0.4875	0.2375	0.0000	0.86
HiSynHiSem	0.2750	0.2250	0.0000	0.73

Table 2: Accuracy across conditions for the mixed model GPT-2-small attention heads, and human data from Van Dyke (2007).

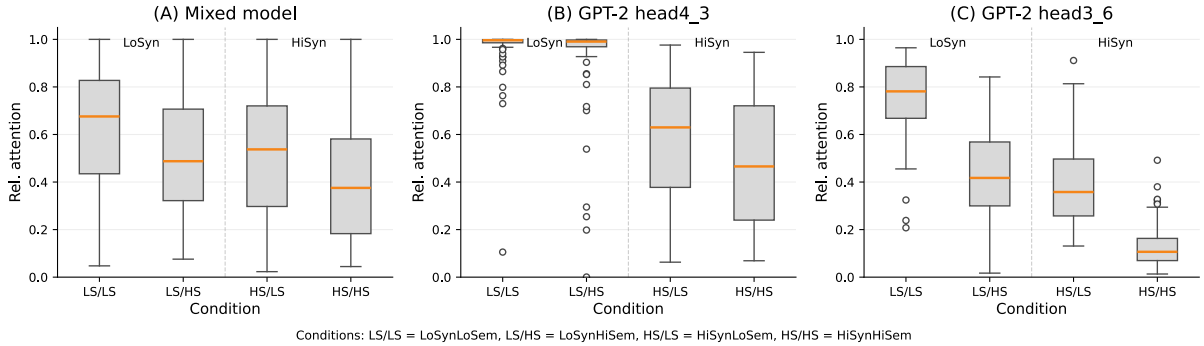


Figure 4: Comparison of relative retrieval score patterns across models and human data. Boxplots display model-predicted relative attention to the correct subject across the four interference conditions from Tan et al. (2017).

icant ($p < 0.05$), consistent with the qualitative human pattern (Van Dyke, 2007).

In contrast, $head4_3$ exhibits near-ceiling relative attention to the correct subject under low syntactic interference and show minimal semantic interference effects, which is consistent with results from Timkey and Linzen (2023). $head3_6$ shows more balanced and separable retrieval-score distributions across all four conditions; however, its low retrieval accuracy indicates that these condition-level effects do not correspond to reliable retrieval of the intended subject.

4.5 Error Analysis

To diagnose this mismatch between condition-level separation and target-level accuracy, we compute each model's average output distribution over six linguistically interpretable regions preceding the verb. The regions include: (1) the determiner of the subject noun phrase, (2) the real subject that should agree with the main verb, (3) introductory elements preceding the subject, (4) the distractor noun within the embedded clause that creates interference, (5) the remainder of the embedded clause following the distractor, and (6) the token immediately preceding the verb (typically the auxiliary "was" in our stimuli). By aggregating attention weights within these syntactic regions, we can better assess whether attention heads are tracking linguistically relevant dependencies or responding to other structural patterns.

Figure 5 shows that $head3_6$ allocates its attention mostly to the auxiliary that immediately precedes it, rather than on the words that are possible as the retrieval cue's intended target. This pattern contrasts sharply with $head4_3$ and our mixed model, which shows stronger attention to the real subjects and distractors. Importantly, when the mixed model makes retrieval errors, these errors tend to favor the interference distractor manipulated in Van Dyke's human experiment rather than linguistically irrelevant tokens, suggesting that the model's failures are themselves aligned with the intended cue-based competition.

This observation raises a broader interpretive question: if a psycholinguistic effect appears in a model component not designed for that function, should we treat it as meaningful evidence of the underlying cognitive mechanism, or does it instead point to a different kind of specialization within the network? This is discussed in greater detail in Section 5.

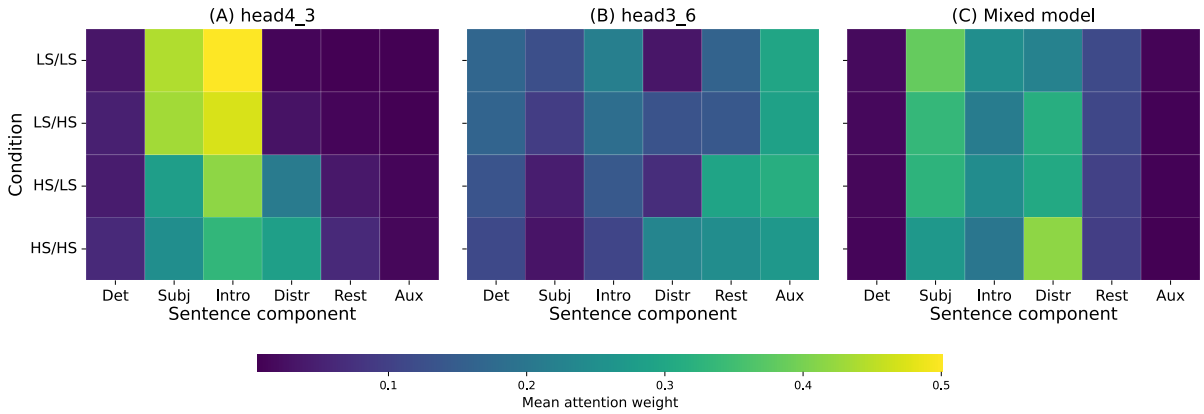


Figure 5: Error analysis from this paper (C) as well as two previously-proposed models of retrieval (A,B). Mean attention weights from the main verb to preceding tokens for mixed model, $head4_3$, and $head3_6$. Rows indicate the syntactic region before the verb cue, columns represent syntactic-semantic interference conditions. Color intensity corresponds to attention weight magnitude (see individual colorbars).

5 Discussion

5.1 Relation to attention-based accounts of retrieval

In our modeling with Tan et al. (2017)'s materials, the attention head proposed by Ryu and Lewis (2021) as syntactically specialized for `nsubj` retrieval ($head4_3$) shows a strong effect of syntactic interference but little sensitivity to semantic interference. This pattern is consistent with the observations of Timkey and Linzen (2023). Rather than contradicting prior work, it suggests that such heads may track structural compatibility cues relevant to subject dependencies while failing to capture the broader combination of syntactic and semantic competition implicated in human interference effects.

At the same time, another head ($head3_6$) exhibits interference-like patterns in its relative attention scores across both syntactic and semantic dimensions. However, its retrieval accuracy is close to zero (Table 2), and the attention-distribution analysis in Figure 5 shows that it allocates most attention to the auxiliary preceding the verb rather than to plausible subject candidates. As noted by Ryu and Lewis (2025), this head is primarily associated with relations such as `nmod:poss` and `nummod`, rather than `nsubj`. The interference pattern therefore arises even though the head's attention is not primarily directed toward the candidates that a subject-retrieval account would identify as relevant. Taken together, these observations highlight a broader challenge for attention-based explanations: neural architectures distribute linguistic computations across many interacting components, and there is currently no foolproof method for identifying which internal units correspond to particular linguistic operations.

5.2 Implications for computational psycholinguistic modeling

These results suggest that relatively simple computational mechanisms may be sufficient to capture key interference phenomena in sentence processing. Syntax and semantics may not be as hopelessly entangled as they appear from the perspective of opaque pretrained neural networks.

More broadly, the findings highlight a trade-off between fit to human data and mechanistic transparency in computational psycholinguistic modeling. In the search for scientific insight into human language comprehension, neural language models can provide strong predictors of human behavior, but the relationship between internal model components and cognitive mechanisms is often difficult to establish. By contrast, explicitly parameterized models such as the one proposed here can transparently identify the source of interference.

From this perspective, the goal of computational psycholinguistics is not only to identify predictors that correlate with human behavior, but also to develop models that clarify the mechanisms generating those effects (Hale, 2017). The present model contributes to this effort by providing a simple formalization of cue-based retrieval in which syntactic and semantic interference arise from independently interpretable computations. Such models can serve as a useful bridge between symbolic accounts of

memory retrieval and distributional representations learned from large corpora.

6 Conclusion

This study presents a computational model of cue-based retrieval interference. By combining these two components within a simple linear retrieval framework, the model captures the graded interference patterns observed in experimental materials manipulating structural similarity and semantic plausibility.

Our comparison with attention-based predictors from GPT-2-small highlights the difficulty of interpreting neural model components as direct implementations of linguistic retrieval mechanisms. While some attention heads exhibit partial interference patterns, their behavior does not consistently align with the retrieval targets predicted by cue-based theory.

More broadly, the results show that key interference effects in sentence processing can be modeled using a transparent and explicitly parameterized architecture. This approach provides a computational bridge between symbolic psycholinguistic theories of memory retrieval and distributional representations learned from large corpora, offering a clearer mechanistic account of how syntactic and semantic similarity interact during comprehension.

Acknowledgments

We thank the anonymous reviewers for their helpful comments and suggestions. We are also grateful to Miloš Stanojević for valuable discussions and feedback on earlier versions of this work.

References

Suhas Arehalli, Brian Dillon, and Tal Linzen. 2022. Syntactic surprisal from neural models predicts, but underestimates, human processing difficulty from syntactic ambiguities. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 301–313, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–265.

Charlotte Caucheteux, Alexandre Gramfort, and Jean-Rémi King. 2022. Deep language algorithms predict semantic comprehension from brain activity. *Scientific Reports*, 12(1):16327.

Christian Clark, Byung-Doh Oh, and William Schuler. 2025. Linear recency bias during training improves transformers' fit to reading times. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7735–7747, Abu Dhabi, UAE. Association for Computational Linguistics.

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.

John Hale. 2017. Models of human sentence comprehension in computational psycholinguistics. In Mark Aronoff, editor, *Oxford Research Encyclopedia of Linguistics*. Oxford University Press.

Philip Hofmeister and Shravan Vasishth. 2014. Distinctiveness and encoding effects in online sentence comprehension. *Frontiers in Psychology*, 5.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. *arXiv preprint arXiv:1902.10186*.

Jungo Kasai, Dan Friedman, Robert Frank, Dragomir Radev, and Owen Rambow. 2019. Syntax-aware neural semantic role labeling with supertags. 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 701–709, Minneapolis, Minnesota. Association for Computational Linguistics.

Mary Alexandria Kelly, Moojan Ghafurian, Robert L West, and David Reitter. 2020. Indirect associations in learning semantic and syntactic lexical relationships. *Journal of Memory and Language*, 115:104153.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland. Association for Computational Linguistics.

Richard L Lewis and Shravan Vasishth. 2005. An activation-based model of sentence processing as skilled memory retrieval. *Cognitive science*, 29(3):375–419.

Richard L. Lewis, Shravan Vasishth, and Julie A. Van Dyke. 2006. Computational principles of working memory in sentence comprehension. *Trends in Cognitive Sciences*, 10(10):447–454.

Bofang Li, Tao Liu, Zhe Zhao, Buzhou Tang, Aleksandr Drozd, Anna Rogers, and Xiaoyong Du. 2017. Investigating different syntactic context types and context representations for learning word embeddings. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages

2421–2431, Copenhagen, Denmark. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Klaus Oberauer and Stephan Lewandowsky. 2008. Forgetting in immediate serial recall: Decay, temporal distinctiveness, or interference? *Psychological review*, 115(3):544.

Byung-Doh Oh and William Schuler. 2022. Entropy- and distance-based predictors from GPT-2 attention patterns predict reading times over and above GPT-2 surprisal. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9324–9334, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Byung-Doh Oh and William Schuler. 2023a. Transformer-based language model surprisal predicts human reading times best with about two billion training tokens. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1915–1921, Singapore. Association for Computational Linguistics.

Byung-Doh Oh and William Schuler. 2023b. 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.

Dan Parker, Michael Shvartsman, and Julie A. Van Dyke. 2017. The cue-based retrieval theory of sentence comprehension: New findings and new challenges. In Linda Escobar, Vicenç Torrens, and Teresa Parodi, editors, *Language Processing and Disorders*, pages 121–144. Cambridge Scholars Publishing, Newcastle.

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

Soo Hyun Ryu and Richard L Lewis. 2025. Memory for prediction: A transformer-based theory of sentence processing. *Journal of Memory and Language*, 145:104670.

Ferdinand de Saussure. 1916. Rapports syntagmatiques et rapports associatifs. *Cours de linguistique générale. Payot, Paris, France*, pages 170–175.

Cory Shain. 2024. Word frequency and predictability dissociate in naturalistic reading. *Open Mind*, 8:177–201.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Miloš Stanojević, Jonathan R Brennan, Donald Dunagan, Mark Steedman, and John T Hale. 2023. Modeling structure-building in the brain with ccg parsing and large language models. *Cognitive science*, 47(7):e13312.

Mark Steedman. 2001. *The syntactic process*. MIT press.

Mark Steedman and Jason Baldridge. 2011. Combinatory categorial grammar. *Non-transformational syntax: Formal and explicit models of grammar*, pages 181–224.

Julius Steuer, Marius Mosbach, and Dietrich Klakow. 2023. Large GPT-like models are bad babies: A closer look at the relationship between linguistic competence and psycholinguistic measures. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 142–157, Singapore. Association for Computational Linguistics.

Yingying Tan, Randi C Martin, and Julie A Van Dyke. 2017. Semantic and syntactic interference in sentence comprehension: A comparison of working memory models. *Frontiers in psychology*, 8:198.

William Timkey and Tal Linzen. 2023. A language model with limited memory capacity captures interference in human sentence processing. *arXiv preprint arXiv:2310.16142*.

Julie A Van Dyke. 2007. Interference effects from grammatically unavailable constituents during sentence processing. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 33(2):407.

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.

Julie A. Van Dyke and Brian McElree. 2006. Retrieval interference in sentence comprehension. *Journal of Memory and Language*, 55(2):157–166.

Julie A. Van Dyke and Brian McElree. 2011. Cue-dependent interference in comprehension. *Journal of Memory and Language*, 65(3):247–263.

Shravan Vasishth and Felix Engelmann. 2021. *Sentence comprehension as a cognitive process: A computational approach*. Cambridge University Press.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head

self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

A Appendix

A.1 Training details

The grammar-bilinear model is trained for 20 epochs using a learning rate of 0.01 without L2 regularization.

For the asymmetric word2vec (Asymw2v) component, we follow the general training configuration of Li et al. (2017). Negative sampling uses 5 negative samples and a distribution smoothing parameter of 0.75. The model is trained for 2 epochs with a learning rate of 0.005 and a batch size of 2048. No dynamic context sampling or subsampling is applied.

The word and context vocabularies include tokens and dependency-labeled contexts with a minimum frequency threshold of 100.

A.2 First word bias

We observe the first word bias in attention heads of GPT-2 Small layer 4, head 3 indicated in (Ryu and Lewis, 2025), as illustrated in Figure 6. In this phenomenon, the attention mechanism disproportionately assigns the highest weight to the initial token of the sequence, regardless of its syntactic or semantic relevance to the current token.

Figure 6: First word bias shown in GPT-2-small $head4_3$

A.3 Ablation of the Grammar-Bilinear Model with CCG Context Windows

We further investigate the impact of the CCG supertag convolution window size k on the model's capacity to identify the correct subject among words preceding a verb. As illustrated in Figure 7, expanding the context window k results in a monotonic increase in prediction accuracy and a concomitant decrease in entropy. The performance gains begin to saturate beyond $k = 4$. This trend confirms that enriching representations with local syntactic history (via concatenated previous supertags) significantly reduces uncertainty and aids in resolving subject-verb dependencies relative to the context-agnostic baseline ($k = 0$). Furthermore, the observed stabilization suggests a theoretical upper bound on the model's performance when using purely local syntactic context.

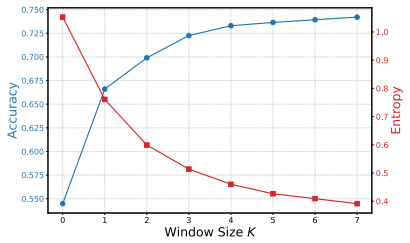


Figure 7: Accuracy and entropy of grammar-bilinear models on the English Web Treebank test set across varying convolution window sizes k. Here, k represents the number of previous tokens' supertags concatenated to the current token's supertag; $k = 0$ denotes a baseline where each token is represented solely by its own supertag without contextual information.

A.4 Probing for Semantic Animacy

To demonstrate that the learned Asymw2v vectors reflect animacy through vector distance in the high-dimensional space, we conduct a linear probe study. We use WordNet to construct a robust animacy lexicon, resulting in a dataset of 9,336 items ($N_{animate} = 4,336$, $N_{inanimate} = 5,000$) after removing OOV. Using these labels, we train a logistic regression classifier as a linear probe on the 300-dimensional frozen embeddings to determine if animacy is linearly separable in the vector space.

As shown in Table 3, the results indicate a strong alignment between the learned vector geometry and semantic animacy. The linear probe achieves an overall accuracy of 91.33% and a ROC-AUC score of 0.9674. The high precision and recall across both classes suggest that the embedding space effectively clusters entities based on agency without explicit semantic supervision.

Figure 8 visualize the separation by projecting test-set embeddings onto the normal vector of the learned decision boundary, yielding a one-dimensional representation along the discrimina-

Metric/Class	Precision	Recall	F1-Score	Support
Inanimate	0.94	0.90	0.92	1000
Animate	0.89	0.93	0.91	868
Overall Accuracy		0.9133		
ROC-AUC		0.9674		

Table 3: Linear probe results for animacy classification using logistic regression on 300-dimensional embeddings. The model demonstrates high separability between animate and inanimate classes.

tive axis. The resulting histogram shows two distinct, largely non-overlapping distributions. The inanimate terms (blue) cluster around a negative mean projection value, while animate terms (orange) cluster around a positive mean. The minimal overlap between the two densities corroborates the high ROC-AUC score reported in Table 3, demonstrating that the semantic distinction between animate and inanimate tokens is geometrically realized linearly in the embedding space.

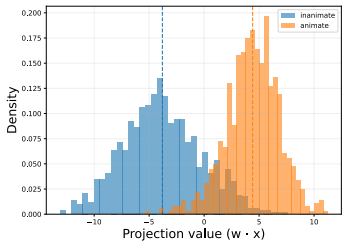


Figure 8: Histogram of projection values ($w \cdot x$) for the test set vocabulary on the learned animacy axis. Dashed lines indicate the mean projection value for each class.

A.5 Model demonstration

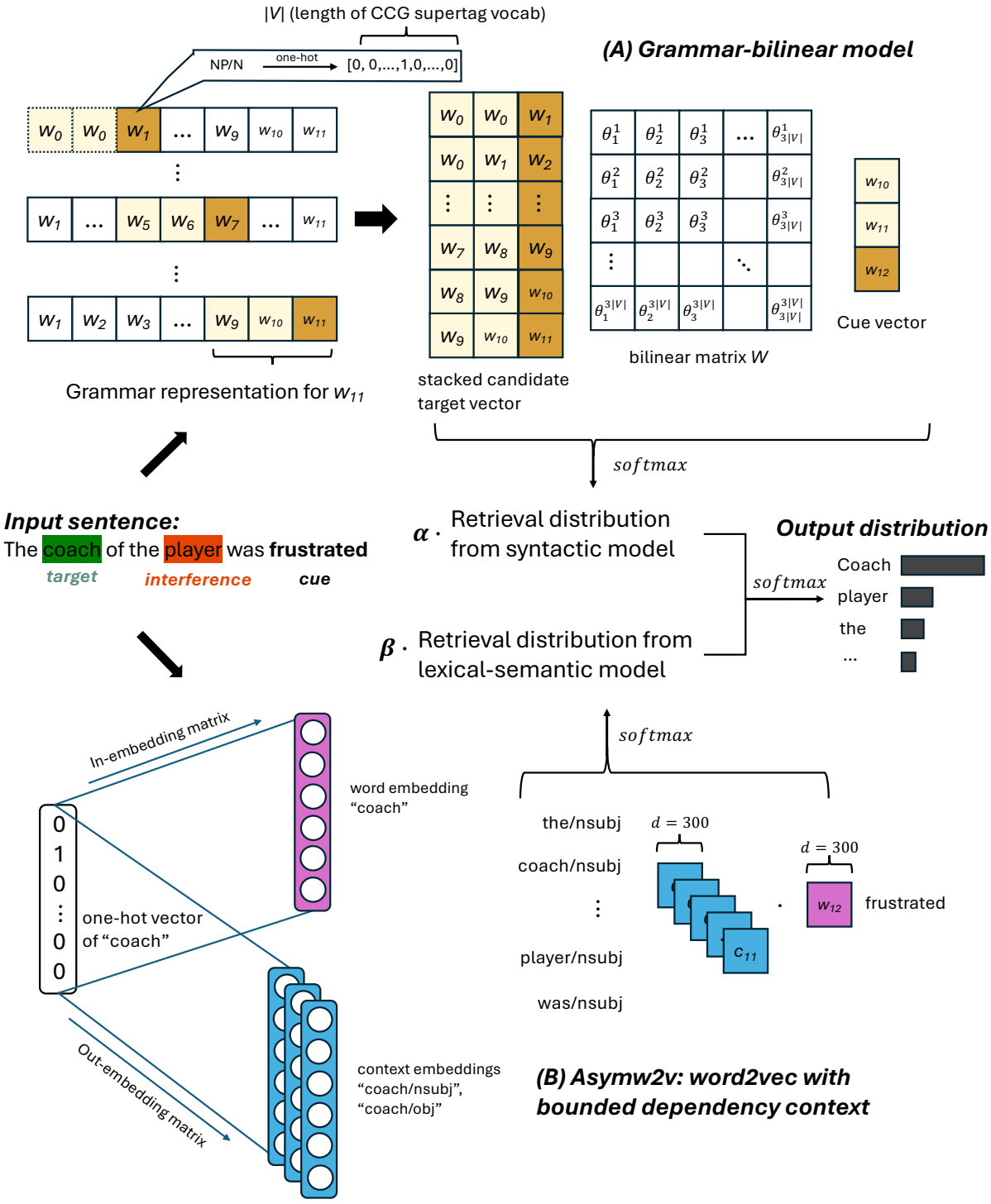


Figure 9: Architecture of the proposed retrieval model. The grammar-bilinear component (A) represents syntactic context using CCG supertags and computes cue–target compatibility through a bilinear transformation to produce a syntactic retrieval distribution. The lexical-semantic component (B) uses dependency-based word embeddings to model semantic compatibility between the cue and candidate targets. The two distributions are combined linearly and normalized to produce the final retrieval probability over candidate subjects.

How much capacity does Turkish inflection require? An empirical study of GRU encoder–decoder bottlenecks

Fred Mailhot
Dialpad, Inc.
fred.mailhot@dialpad.com

1 Introduction

Encoder–decoder neural networks with high-dimensional embeddings and hidden layers (typically d=300–500) are now the default tool for modeling morphophonological phenomena as sequence-to-sequence mappings, achieving high accuracy across languages and pattern types (cf. Kirov and Cotterell, 2018; Corkery et al., 2019; Mayer and Nelson, 2020; Prickett, 2021; Li et al., 2024, *inter alia*). For interpretability and cognitive-modeling purposes, this raises an obvious but under-examined question: how much of that capacity is necessary? Earlier connectionist work on similar problems succeeded with networks of single-digit hidden dimensionality (e.g. Rodd, 1997, for vowel harmony), raising the possibility that contemporary models may be substantially overparameterized, at least for some morphophonological tasks.

We address this question here for the case of Turkish morphological inflection by training very small GRU encoder–decoder models *without* attention, sweeping the dimensionality of an enforced bottleneck between the encoder and decoder. The attention-free design is deliberate: the bottleneck enforces a strict constraint on the information that propagates from encoder to decoder, so the question "how much capacity is needed?" becomes a measurable property of the data and task rather than of the architecture.

We report three findings: (i) sequence accuracy follows a sigmoid shape in bottleneck dimension, with a transition zone between h=16 and h=48, with performance near ceiling at h=64 (∼70k parameters), (ii) logistic regression probes on the bottleneck recover the harmony class [±back] of stem vowels at $\geq 95\%$ accuracy across *all* sizes, including sizes at which sequence accuracy is near zero, (iii) adding Bahdanau attention does not eliminate the bottleneck's role: sequence accuracy for attention without context transmission collapses to $\sim 1\%$, producing harmonically fluent but morphologically wrong outputs. Together, these results give a tight, replicable lower bound on the capacity required for this task in this architecture, and a clean picture of what the bottleneck encodes *before* it can support the full inflection task.

2 Methods

2.1 Data

We use the Turkish dataset from the CoNLL–SIGMORPHON 2018 Shared Task ($10k$ train, $1k$ dev). Inputs are *(lemma, feature-bundle)* pairs in Turkish orthography (e.g. *artık*, `N;GEN;PL`) and outputs are inflected surface forms (e.g. *artıkların*).

2.2 Model

Our model has a two-part encoder: a bidirectional GRU that encodes the input stem character sequence, and a pooled-embedding (d=8) MLP that encodes the morphosyntactic feature bundle. The stem encoder's final hidden state and the feature representation are concatenated and linearly projected (with *tanh* activation) to an h-dimensional *bottleneck context* vector. A unidirectional GRU decoder generates the output character-by-character, receiving the context vector concatenated to the character embedding at every timestep (teacher forcing during training; greedy decoding at inference). Stem encoder and decoder share (tied) d=8 character embeddings.[1]

2.3 Training

We train with AdamW: lr=10^{-3}, weight decay=10^{-3}, `ReduceLROnPlateau` scheduling, batch size=32, gradient clipping=1.0, and early stopping (patience=15 on validation loss).

[1] Character embedding dimension was swept from d=3 to d=32; performance exhibited a sharp transition at d=8, which we adopted as the fixed embedding size for the bottleneck sweep reported here.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 423–426
July 3-4, 2026 ©2026 Association for Computational Linguistics

Models are trained on a Google Cloud VM with a single T4 GPU for a maximum of 400 epochs.

2.4 Bottleneck sweep

The bottleneck dimension h is swept across {8, 12, 16, 20, 24, 32, 48, 64, 96, 128} with five random seeds each (50 runs total). Model sizes range from $\sim$2.6–95k parameters.

2.5 Probing

To test whether the bottleneck encodes phonological structure, we extract context vectors from trained models and train logistic regression probes to predict the backness class ([±back]) of each stem's final vowel. We evaluate probe accuracy across all 10 hidden layer sizes. As controls—and to test whether the bottleneck encodes the features needed for Turkish's secondary rounding-harmony pattern—we train analogous probes for the rounding class ([±round]) and the binary height ([±high]) of the same vowel.

2.6 Attention ablation

To confirm the bottleneck is a genuine constraint rather than a data size or training artifact, we add Bahdanau attention and test three decoder configurations: (a) attention only (no context vector), (b) attention with context as decoder initialization (INIT_CTX), and (c) attention with context concatenated at every step (CONCAT_CTX). Each is swept over $h \in \{16, 20, 24, 32, 48\}$ with 5 random seeds.

3 Results

3.1 Sequence accuracy

Mean sequence accuracy (Figure 1) across hidden dimensions reveals a sigmoid-shaped curve, near-zero at h=8 (0.6%), rising steeply through the transition zone (h=16–48), and approaching ceiling by h=64 (92.0%). Variance peaks in the transition zone (h=24: σ=3.3) and is minimal at both extremes, consistent with a regime where small differences in initialization determine whether the model finds a good solution.

3.2 Harmony-aware edit distance

Exact-match sequence accuracy is a coarse signal: a model that fails to produce the right suffix may still produce a vowel of the correct harmonic class. We complement seq. acc. with a harmony-aware edit distance, in which vowel substitutions are weighted by phonological feature distance (backness and rounding) rather than treated as unit edits. This metric decreases monotonically in h from 6.5 at h=8 to 0.15 by h=96 — mirroring the sigmoid in seq. acc. but with no visible threshold. Even sub-ceiling models produce forms increasingly close to the target in feature space, consistent with the probing picture below.

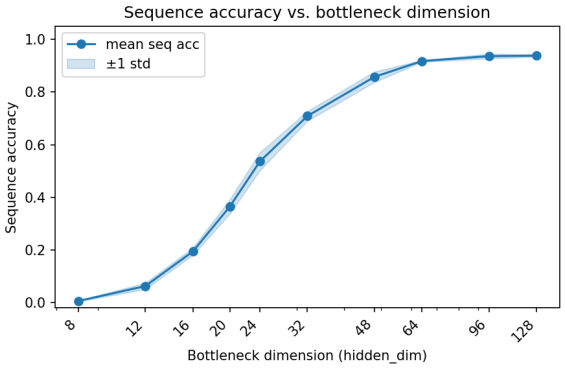


Figure 1: Sequence accuracy on dev set (mean ± std across 5 seeds) as a function of bottleneck dimension h.

h	[±back]	[±round]	[±high]	Seq. acc.
baseline	52.9	84.5	66.1	—
8	97.8	86.9	72.1	0.6
16	96.1	90.9	83.5	19.5
24	98.6	95.0	92.8	53.6
48	99.7	97.9	97.3	85.9
128	99.9	99.1	97.5	93.8

Table 1: Probe and sequence accuracy (%) vs. bottleneck dimension h. Probe columns predict [±back], [±round], and [±high] of the stem's final vowel; the *baseline* row gives majority-class accuracy on the training set (weighted by inflection frequency). [±back] is encoded well above baseline even at h=8, where inflection accuracy is near zero. [±round] and [±high] are both close to their respective baselines at low capacity and only diverge meaningfully at larger h.

3.3 Probing

Logistic-regression probes for [±back] achieve 96.1–99.9% accuracy across all bottleneck sizes (Table 1) – roughly 45 points above the 52.9% majority-class baseline. The dissociation at h=8 is striking: the model cannot produce correct outputs at all (0.6% seq. acc.), yet a linear classifier reads the stem's harmony class off the context vector essentially perfectly. *The bottleneck encodes what class the stem belongs to before it can encode enough to produce the right output.*

The dissociation does not extend to other vowel features. [±round] and [±high] have substantially higher majority-class baselines (84.5% and 66.1%

respectively), reflecting Turkish's phonotactic restriction on round vowels outside the first syllable and the lexical dominance of non-high vowels in stem-final position. Once baselines are accounted for, [±round] sits only 2.4 pp above baseline at h=8, and [±high] only 6.0 pp; both probes are essentially at majority-class performance at low capacity. By h=48 they rise to +13.4 pp and +31.2 pp above baseline respectively.[2]

One might worry that [±back] decoding is so successful at h=8 simply because Turkish orthography makes front/back vowel identity directly readable from individual characters—that the probe is recovering a surface graphemic property rather than harmonic structure. But Turkish orthography is transparent for *all three* vowel features we test: [±back], [±round], and [±high] are each fully determined by individual vowel characters. If trivial orthographic decoding were the explanation, all three should be equally recoverable above their respective baselines. They are not—only [±back] is.

The picture is therefore narrower and sharper than a generic "phonology before morphology" claim: the bottleneck does not encode vowel features in general at low capacity; it encodes the *one* binary distinction Turkish harmony is built on. [±back] is the feature the decoder must propagate to choose between front- and back-harmonic suffix allomorphs; encoding it is necessary, though not sufficient, for the full task. The features required only for the conditioned secondary rounding pattern emerge only with capacity.

3.4 Attention ablation

Adding attention *without* the context vector is catastrophic (Table 2): accuracy drops to ∼1% at all hidden dimensions. These models produce fluent, harmonically consistent Turkish—but with wrong suffixes, because the decoder never receives the morphosyntactic feature specification. Notably, harmony-aware edit distance for these models is *lower* than for the smallest baseline models (∼5.5 vs. 6.5 at h=8), confirming that they produce harmonically well-formed but morphologically incorrect output.

Restoring context access recovers performance. Initializing the decoder hidden state from the context vector (INIT_CTX) partially restores the capacity curve (31.6% at h=16, 94.7% at h=48), but with high variance at small sizes. Concatenating the context at every step (CONCAT_CTX) flattens the curve: even h=16 models reach 84.7%, and performance is near-ceiling by h=24 (92.5%).

Condition	h=16	h=20	h=24	h=32	h=48
No attention	19.5	36.6	53.6	70.9	85.9
Attn, no ctx	1.1	1.4	1.3	1.0	1.5
Attn+init_ctx	31.6	65.2	79.5	91.3	94.7
Attn+concat_ctx	84.7	90.3	92.5	94.5	95.8

Table 2: Sequence accuracy (%) across attention conditions. "No attention" is the baseline from Figure 1. All values are means over 5 seeds.

The contrast between INIT_CTX and CONCAT_CTX is informative. When the context is only available at initialization, it degrades over the course of generation—the decoder must maintain feature information in its recurrent state while simultaneously tracking the output sequence. When context is available at every step, this burden is removed, and the attention mechanism's ability to look up stem characters provides the remaining information needed for inflection.

4 Discussion

4.1 What capacity is "enough"?

The sequence-accuracy curve has a clear transition zone (h=16–48) and reaches ceiling by h≈64. We emphasize that this is a lower bound for *this architecture* on *this task and dataset*: a GRU encoder–decoder without attention, trained on the 10k-example CoNLL–SIGMORPHON 2018 Turkish split. We do not claim that 64 hidden units is the universal capacity floor for Turkish inflection—only that, under standard training, more capacity than this is unnecessary, and meaningfully less is insufficient.

4.2 A property of the data, not (just) the network

The dissociation between probe accuracy and task accuracy invites a learnability reading. A bottleneck of h=8 has too little capacity to support correct inflection, but enough to carry a binary distinction over the input; and the binary distinction the model finds first is exactly the one Turkish harmony is built on. We read this as a claim about the input distribution rather than about a particular network: in a corpus of Turkish word forms encoded as orthographic strings, the [±back] partition

[2] Probes use the same logistic-regression setup for all three features; only the target label changes. [±high] is binarised as high vs. mid/low.

of the lexicon is the single most informative feature for predicting suffix shape, and it is the easiest to recover from finite samples even under severe representational constraint. The relevant caveat is that orthography is not raw signal: Turkish orthography is near-phonemic, so character-level inputs already deliver vowel features almost without distortion. Whether the same pattern would hold for a model trained on speech or a less transparent script is a separate question.

4.3 Attention does not erase the bottleneck

The attention ablation shows that allowing the decoder to attend over stem characters is not a substitute for transmitting the morphosyntactic feature bundle. Attention-only models reach $\sim 1\%$ accuracy at every h tested, while producing harmonically well-formed Turkish (mean harmony-edit distance ≈ 5.5, vs. 6.5 for the smallest no-attention baseline). When the bottleneck is restored—even just as decoder initialization—accuracy rises with h in the familiar way. Reinjecting context at every step (CONCAT_CTX) flattens the capacity curve almost entirely. The bottleneck's load-bearing role is the morphosyntactic feature bundle, not stem phonology per se: phonology can be re-fetched via attention; the feature specification cannot.

5 Scope and limitations

Our results are limited along three axes. *Architecture*: we report numbers for GRU encoder–decoders only; LSTM, Transformer, and convolutional variants may have different capacity profiles. *Language*: Turkish is a typologically agglutinative language with transparent (orthographically marked) vowel harmony; we expect quantitatively different curves for fusional or polysynthetic systems, and for harmony systems with transparent or neutral vowels (e.g. Finnish). *Data scale*: the SIGMORPHON 2018 *high* split is small relative to modern training corpora; minimum-capacity thresholds may shift under different data regimes. We therefore present our central quantitative claims (transition zone, ceiling capacity) as architecture- and corpus-specific empirical lower bounds, not as universal capacity floors.

6 Future work

The most immediate extension is cross-linguistic: applying the same bottleneck sweep to Finnish (which has transparent vowels and a richer harmony inventory) would test whether the capacity threshold for transparent-vowel harmony is meaningfully higher than for Turkish's opaque system. More broadly, comparable sweeps on a fusional language (e.g. Russian) and a polysynthetic one (e.g. Yupik) would calibrate whether the "small models suffice" finding generalises beyond agglutinative morphology, or whether it depends on the comparatively local conditioning of Turkish inflection. Finally, the dissociation between [±back] encoding and [±round]/[±high] non-encoding we observe at low capacity is a candidate experimental probe for cognitive learnability claims; replicating it under inputs that are not orthographically near-phonemic (raw audio features, or non-transparent scripts) would test how much of the effect we attribute to the data is in fact an artefact of the input encoding.

References

Maria Corkery, Yevgen Matusevych, and Sharon Goldwater. 2019. Are we there yet? encoder-decoder neural networks as cognitive models of English past tense inflection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3868–3877, Florence, Italy. Association for Computational Linguistics.

Christo Kirov and Ryan Cotterell. 2018. Recurrent neural networks in linguistic theory: Revisiting Pinker and Prince (1988) and the past tense debate. *Transactions of the Association for Computational Linguistics*, 6:651–665.

Jane Li, Kyle Rawlins, and Paul Smolensky. 2024. What representations do rnns learn and use from morpho-phonological processes? In *Society for Computation in Linguistics*, volume 7, page 289–290.

Connor Mayer and Max Nelson. 2020. Phonotactic learning with neural language models. In *Society for Computation in Linguistics*, volume 3, pages 291–301.

Brandon Prickett. 2021. *Learning Phonology with Sequence-to-Sequence Neural Networks*. Ph.D. thesis, University of Massachusetts Amherst.

Jennifer Rodd. 1997. Recurrent neural-network learning of phonological regularities in Turkish. In *CoNLL97: Computational Natural Language Learning*.

The signal is coming from inside the noun phrase! Tracking semantic proto-role inferences during sentence processing

Lucas Y. Li
Cornell University
lyl27@cornell.edu

Zander Lynch
Cornell University
zcl7@cornell.edu

Marten van Schijndel
Cornell University
mv443@cornell.edu

Abstract

Semantic roles between a predicate and argument can be decomposed into proto-role properties (e.g., INSTIGATION). We introduce a novel LLM feature attribution method, Generalized Contextual Decomposition for Transformers (GCD-T), which we use to probe which parts of a sentence enable models to infer proto-role properties. We compare our findings with human inferences.

1 Introduction

Semantic roles characterize the relationship between entities and their predicates. For example, agents are initiators of events while patients receive event outcomes. These broad roles can be decomposed into combinations of specific semantic features called *proto-role properties* (Dowty, 1991), which are categorized by whether they reflect more agent-like or patient-like features. When processing a sentence, one must infer these proto-role properties to form an understanding of the events being conveyed (who caused the event and how were others impacted?). However, it remains an open question which parts of a sentence convey these properties to readers.

Various linguistic theories have posited different lexical categories to project proto-role properties but these theories have not been tested against what processors actually use to infer these properties in a sentence. Following Jackendoff (1976) and Davidson (1967), Dowty (1989) saw semantic roles and thus proto-role properties (PRPs) as "determined completely and solely by verb meanings." For example under this view, x *murders* y entails that x is VOLITIONAL no matter the identity of x, while x *kills* y does not, and both entail y is SENTIENT. Other scholars argue that semantic roles are also determined by the semantic properties of the arguments or the syntax (Parsons, 1990; Kratzer, 1996; Williams, 2015; Husband, 2023).

Even if semantic roles *are* fully specified by the verb, arguments may still be required to carry the relevant PRPs in order to be validated against the argument slots entailed by the verb. This explains why (1a) is acceptable while (1c-d) are not and (1b) is only allowed metaphorically, as *the disease* cannot be SENTIENT nor VOLITIONAL.

(1) a. The disease killed Mary.
b. ? The disease murdered Mary.
c. # Mary killed the disease.
d. # Mary murdered the disease.

Intuitively, we know that *Mary* is SENTIENT and *the disease* is not, irrespective of whether this is also lexically entailed by the verb. Therefore, our research question is *how much* of the signal used to predict PRPs comes from the argument compared to the verb, or the other words in the sentence? In this work, we introduce a novel analytical method to probe the contribution of nouns, modifiers, and verbs to PRP prediction in LLMs and validate our findings with an offline judgment task by humans. While we do not assert that LLMs model human cognition, they learn *statistical regularities* over their training corpora and thus can reveal what information is useful for accurate semantic inference. We find that both LLMs and human annotators utilize nouns and adjectives in addition to verbs to infer proto-role properties.[1] We list our major contributions below:

- We implement Generalized Contextual Decomposition for Transformers (GCD-T) to quantify the contributions of arguments, modifiers, and predicates in the Semantic Proto-Role Labelling (SPRL) task.
- We find that NPs are at least as informative for SPRL as verbs, especially for grammatical

[1]The code for our experiments can be found at https://github.com/lucas-y-li/decomposing_transformers

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 427–441
July 3-4, 2026 ©2026 Association for Computational Linguistics

agents (active subjects and passive objects) and agentive properties.

- We confirm experimentally that these computational results generalize to humans. We find that both nouns and adjectives significantly affect human PRP judgments of agentive properties, indicating that humans use non-verb elements when inferring semantic role properties.

2 Background

2.1 Proto-Role Property Labelling

PRP judgments have been experimentally studied using likelihood ratings of questions that accessibly paraphrase each property (Kako, 2006). For example, to elicit a rating for VOLITION, participants were asked, "How likely is it that the [Arg] chose to be involved in the [Pred]?" Reisinger et al. (2015) and White et al. (2016) later scaled these elicitations to produce a large-scale database of crowdsourced annotations, which they released as the Universal Decompositional Semantics dataset (Decomp).[2]

Following the introduction of Decomp, several works have presented computational models for the Semantic Proto-Role Labelling (SPRL) task, which aims to predict the likelihood of each PRP for an argument-predicate pair in a sentence. Teichert et al. (2017) first approached the task as a binary multi-label classification problem using a Conditional Random Field (CRF) model. Similarly, Rudinger et al. (2018) proposed a bidirectional long-short term memory (BiLSTM) model trained with multi-task learning and found that predicting all PRPs jointly resulted in effective parameter sharing. Our work is the first to probe SPRL classifiers to investigate which tokens in the input the models utilize when making decisions.

2.2 Pre-Verbal Semantic Role Prediction

Existing psycholinguistic research has found cross-linguistic evidence that humans use semantic information from pre-verbal NPs to infer semantic roles while reading sentences, prior to seeing the verb (Bornkessel et al., 2003; Chow et al., 2018; Liao et al., 2022). These studies argue that, based on an incremental processing model, humans must make preliminary predictions about the semantic roles of pre-verbal NPs using incomplete information. However, the influence of post-verbal NPs on semantic role inference, when the complete sentence is available to the readers, is not yet known (Husband, 2023). These studies were also limited in that they only tested small sets of controlled stimuli. Our computational approach allows the comparison of event elements in a large corpus of naturalistic sentences.

Property Type	Proto-Role Properties
Proto-Agent	AWARENESS, SENTIENT, VOLITION, INSTIGATION, EXISTEDBEFORE, EXISTEDDURING, EXISTEDAFTER
Proto-Patient	CHANGEOFSTATE, CHANGEOFLOCATION, CHANGEOFPOSSESSION

Table 1: Proto-role properties used in our analysis.[3]

3 Interpretability Analysis Method

3.1 Generalized Contextual Decomposition

To quantify the degree to which the models utilize NPs and verbs during SPRL, we implement Generalized Contextual Decomposition (GCD), an interpretability method originally proposed for LSTM gates (Murdoch et al., 2018; Jumelet et al., 2019), which we further generalize to self-attention to apply it to Transformer-based models.

Given an input sequence $\mathbf{x}$, GCD partitions the tokens into *in-focus* (β) and *out-of-focus* (γ) components, $\mathbf{x}_\beta$ and $\mathbf{x}_\gamma$. This is done through masking, such that $\mathbf{x} = \mathbf{x}_\beta + \mathbf{x}_\gamma$, and each component has zero embeddings for the tokens not assigned to it. The bias component $\mathbf{x}_\delta$ is initialized as zeros. The goal of GCD is to decompose the output of encoder $\mathcal{E}$ into its constituent contributions $L_\mathcal{E}$,

$$\mathcal{E}(\mathbf{x}) = L_\mathcal{E}(\mathbf{x}_\beta) + L_\mathcal{E}(\mathbf{x}_\gamma) + L_\mathcal{E}(\mathbf{x}_\delta) \quad (1)$$

This is accomplished by propagating the in-focus and out-of-focus partitions through all layers of $\mathcal{E}$. For any layer ℓ, its hidden state representation $\mathbf{h}^\ell$ is linearly decomposed into

$$\begin{aligned}\mathbf{h}^\ell &= L_\ell(\mathbf{h}_\beta^{\ell-1}) + L_\ell(\mathbf{h}_\gamma^{\ell-1}) + L_\ell(\mathbf{h}_\delta^{\ell-1}) \\ &= \mathbf{h}_\beta^\ell + \mathbf{h}_\gamma^\ell + \mathbf{h}_\delta^\ell\end{aligned} \quad (2)$$

where $\mathbf{h}_\beta^\ell$ contains the information the layer received from the in-focus input, $\mathbf{h}_\gamma^\ell$ contains information from the out-of-focus input, and $\mathbf{h}_\delta^\ell$ contains information from the bias parameters of layers $\leq \ell$. Thus, GCD reveals how much, and whether positively or negatively, each component *contributes to* each classifier logit. This is distinct from model interpretability methods that analyze

[2] https://decomp.io/projects/semantic-proto-roles

[3] CHANGEOFLOCATION is sometimes considered a proto-agent property. See §4.4 for more discussion of why we consider it a proto-patient property in the Decomp dataset.

attention weights, which instead reveal the information a model has *access to* for classification.

The output of a fully-connected layer can be easily linearly decomposed,

$$L_{FC}(\mathbf{x}_i) = \mathbf{W}^T\mathbf{x}_i, \;\; i \in \{\beta, \gamma\}$$
$$L_{FC}(\mathbf{x}_\delta) = \mathbf{W}^T\mathbf{x}_\delta + \mathbf{b} \quad (3)$$

Non-linear activation functions are decomposed using Shapley values (Shapley, 1953).

3.2 GCD for Transformers

In this work, we introduce Generalized Contextual Decomposition for Transformers (GCD-T). We follow Jumelet et al. (2019) in allowing out-of-focus material to inform decisions about *when* to use in-focus information (and vice-versa), but the representations themselves are kept in isolated in-focus and out-of-focus partitions.[4]

Vaswani et al. (2017) defines self-attention as

$$Attn(\mathbf{x}) = \mathbb{A}(\mathbf{x})\mathbf{V}, \;\; \mathbb{A}(\mathbf{x}) = \text{Softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}),$$
$$\mathbf{Q} = \mathbf{W}_Q^T\mathbf{x}, \; \mathbf{K} = \mathbf{W}_K^T\mathbf{x}, \; \mathbf{V} = \mathbf{W}_V^T\mathbf{x} \quad (4)$$

Where $\mathbb{A}(\mathbf{x})$ is the attention weights, and $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are the queries, keys, and values, respectively. We do not decompose $\mathbb{A}(\mathbf{x})$ so the layer receives the full context to determine which $\mathbf{x}_\beta$ values to propagate forward.$\mathbf{V}$ is decomposed linearly as normal. Thus, the decomposition is

$$L_{Attn}(\mathbf{x}_i) = \mathbb{A}(\textstyle\sum_i \mathbf{x}_i)\mathbf{W}_V^T\mathbf{x}_i, \; i \in \{\beta, \gamma, \delta\} \quad (5)$$

Ultimately, we want to determine the contributions of the in-focus partition to the classification logits ($\mathbf{z}_\beta$). While the original Murdoch et al. (2018) CD method used raw output logits ($\hat{\mathbf{z}}_\beta$), we normalize the raw outputs by the token length of the in-focus element ($|\beta|$): $\mathbf{z}_\beta = \hat{\mathbf{z}}_\beta / |\beta|$. These per-token contributions prevent long β spans (such as complex NPs) from having disproportionately large contributions.[5]

4 Semantic Proto-Role Inference

We now turn to our main research question: investigating which tokens give rise to PRP inferences during processing.

[4]This differs from Murdoch et al., who fully partition representations across all layers. Further discussion about the differences between GCD and CD and an empirical analysis with respect to Transformers is given in §B.

[5]An empirical validation of GCD-T by comparing behaviour to GCD on the tasks of subject-verb number agreement and sentiment analysis is shown in §C.

4.1 Dataset

Following prior work on SPRL, we use the Decomp dataset (Reisinger et al., 2015; White et al., 2016). The dataset consists of 15,000 PRP judgments across 10 properties, shown in Table 1. Annotators used a 5-point likert scale to judge the appropriateness of a PRP applying to a sentence (see §2.1). As is standard with this data, we binarized the judgments into ≤ 3 and > 3. Full preprocessing details and examples are given in §A.

4.2 Models

We evaluate the Transformer models RoBERTa-large (355M parameters; Liu et al., 2019) and GPT-2-medium (345M parameters; Radford et al., 2019). The models are the same size, however RoBERTa is encoder-only and has bidirectional context while GPT-2 is autoregressive and thus must base its decisions only on left context, analogously to humans during incremental processing.[6] We fine-tune the LLMs for the binary semantic proto-role labelling (SPRL) task on Decomp.[7] We insert custom special tokens <|*pred*|> and <|*arg*|> around the relevant predicate and argument of each sentence to indicate which argument-predicate pair is being queried.

Evaluation shows that both models perform comparably to average human inter-annotator agreement on all properties (Table 2).[8] There is no notable difference in F1 scores between active and passive sentences nor between models. Although GCD-T can be applied to any large Transformer LLM with trivial modifications, we choose to probe RoBERTa-large and GPT-2-medium as they already perform comparably to humans on the SPRL task despite their small size.

4.3 Experimental Setup

We partition each sentence-argument-predicate triplet into the following in-focus token spans: ARG, which contains the tokens of the target argument excluding any modifiers; MOD, the modifier tokens of the target argument; PRED, the tokens of the predicate verb; and OTHER, which consists of all non-target arguments of the predicate, if any.

[6]We also experimented with the smaller models RoBERTa-base, GPT-2-small, DistilGPT, and DistilRoBERTa (Sanh et al., 2019) and found similar results.

[7]Both models were fine-tuned with hyperparameters {*lr=5e-5, λ=0.1, epochs=4, batch_size=16, dropout=0.1*}.

[8]Each Decomp V2 item has two annotator ratings, so human F1 is calculated as pairwise micro-F1. Model F1 is calculated as average micro-F1 with respect to both annotations.

Property	Humans		RoBERTa		GPT-2	
	ACT	PASS	ACT	PASS	ACT	PASS
awareness	89.7	88.9	**91.7**	89.7	91.2	**90.5**
Δ_location	77.9	**77.8**	**80.0**	77.4	81.8	79.0
Δ_state	65.0	**68.3**	**67.2**	65.5	**69.3**	68.3
Δ_possession	91.3	90.1	**92.8**	**91.1**	92.5	90.7
existed_after	85.5	86.9	**87.5**	**88.3**	87.2	87.5
existed_before	85.0	85.7	**86.7**	**86.9**	86.4	84.1
existed_during	96.1	96.4	**97.2**	**97.4**	96.7	97.8
instigation	72.4	69.0	**75.7**	**71.4**	73.0	68.3
sentient	89.5	88.5	**92.1**	**90.3**	92.6	91.5
volition	86.8	88.1	**88.2**	**88.5**	86.3	87.7
average	83.9	84.0	**85.9**	**84.6**	85.7	84.5

Table 2: Decomp inter-annotator and model F1.

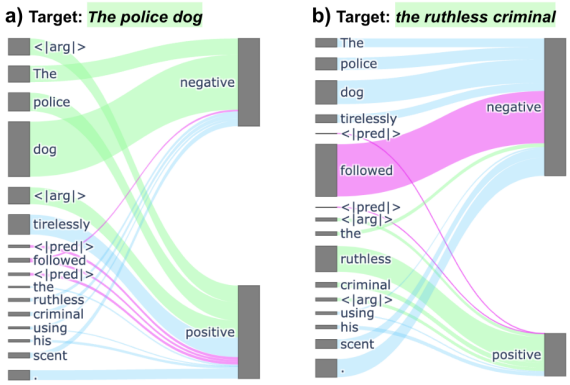


Figure 1: Example SPRL GCD-T GPT-2 token contributions for INSTIGATION. Target NP (MOD, ARG) contributions are shown in green and PRED contributions are in pink.

Argument modifiers are obtained using the gold-label dependency relations and syntactic parses from Universal Dependencies (Silveira et al., 2014). We recursively consider any optional adjunct and the nodes it dominates to be modifiers, unless the argument itself is an adjunct (e.g., PP), in which case we start with its children.

GCD-T is applied to obtain each component's contribution to each PRP. We first show an example decomposition before presenting a quantitative analysis of our Decomp V2 test set.

4.4 GCD-T Results

Qualitative Example. Figure 1 shows an example GPT-2 decomposition for the proto-agent property INSTIGATION of the sentence "The police dog tirelessly followed the ruthless criminal using his scent." In Figure 1a, the target argument is *The police dog* and in Figure 1b, the target argument is *the ruthless criminal*. Both arguments share the *followed* predicate. Based on our categorization, *ruthless* is considered a modifier, but *police* is not because *police dog* is a compound noun.

In Figure 1a, GPT-2 identifies *dogs* as typically non-instigative, which is correct given the training data as non-human living entities are less likely to initiate actions than humans. However, the attribute *police* is identified as contributing positively to instigation. In Figure 1b, the model identifies objects of *followed* as typically non-instigative and assigns it a negative contribution, while the modifier *ruthless* contributes positively.

Quantitative Results. First, we observe that although Reisinger et al. (2015) described CHANGEOFLOCATION as a proto-agent property, GCD-T reveals that it behaves more similarly to proto-patient properties[9] and we thus consider it as such. This categorization is supported by Dowty (1991), who states that CHANGEOFLOCATION is agentive only if the other argument is non-instigative, otherwise it is patientive. As annotators labeled each PRP in Decomp separately, they would not have been aware of this detail, so in this way we can see that GCD-T actually provides empirical evidence for a theoretical PRP distinction.

For proto-agent properties (Figure 2a, 2c), the ARG rarely contributes negative evidence, and only contributes positive evidence in active subjects or passive objects (grammatical Agents). In RoBERTa, ARG and PRED contribute equally (Figure 2a), however GPT-2 disproportionately identifies agentiveness using the ARG (Figure 2c). Surprisingly, passivization has only a small effect on ARG contributions, as contribution trends are largely the same for active subjects and passive objects (grammatical Agents), and active objects and passive subjects (grammatical Patients). Therefore, it appears there is little difference in how models use the PRP cues of post-verbal NPs compared with pre-verbal NPs, even for the incremental model GPT-2.

For proto-patient properties, ARG and PRED contribute largely negatively in RoBERTa (Figure 2b) but positively in GPT-2 (Figure 2d). Relative to ARG, PRED contributes more to GPT-2's proto-patient interpretation than its proto-agent interpretation, particularly in active sentences, indicating that agentive inferences primarily originate from arguments, while patienthood primarily originates from verbs with some input from entities. This is consistent with theoretical proposals that agents,

[9] CHANGEOFLOCATION contributions have mean KL divergence of 6.9 from proto-agent properties and 2.5 from proto-patient properties in RoBERTa and 5.7 from proto-agent properties and 4.5 from proto-patient properties in GPT-2.

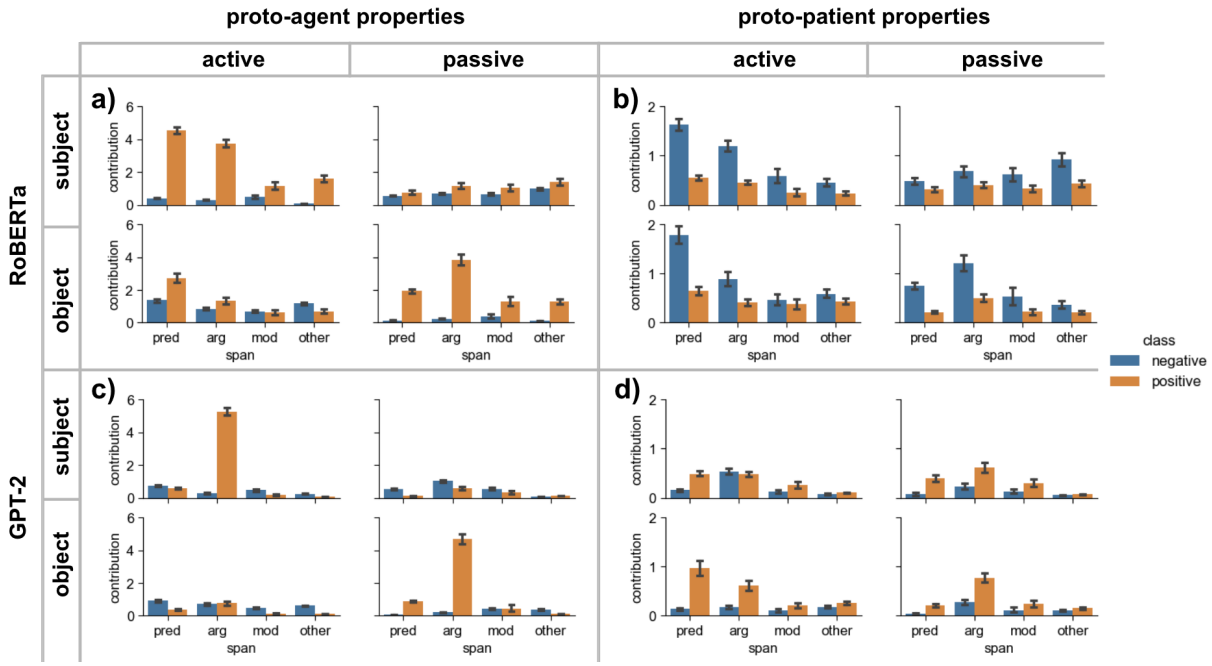


Figure 2: Average GCD-T SPRL token contributions of event constituents. Scales differ between proto-agent and proto-patient contributions likely because of the greater frequency of proto-agent properties in the training data (e.g., in intransitive sentences). Error bars represent 95% CI.

but not patients, are severed from the verb as the verb combines directly with the latter and not the former (e.g., Kratzer, 1996; Levin, 2012).

While GPT-2 identifies positive evidence for both proto-agent and proto-patient properties (Figure 2c, 2d), RoBERTa focuses mostly on positive evidence of *agentiveness* even when the goal of the task is to identify a proto-patient property (Figure 2a, 2b). This perhaps indicates the strength of proto-agentive features learned by RoBERTa during pretraining. Furthermore, although MOD and OTHER contribute less to SPRL than ARG and MOD, their contributions are still non-zero. This is particularly interesting for OTHER, as it means semantic role properties can depend on other sentence elements *outside* of the target predicate-argument pair. Overall, it is clear that both models strongly utilize non-predicate elements for SPRL of all PRPs and syntactic positions.

4.5 Contextualization Analysis

A possible algorithmic hypothesis about what causes the above results is that, rather than relying on ARG cues to predict PRPs (as the above results suggest), the models could rely on ARG to identify the word sense of PRED, and once the correct sense is selected, the models use that verb sense to infer the PRPs. Because GCD-T isolates ARG from PRED, the results look like the models use ARG because that is the first step in their processing pipeline, when in fact they may rely on the sense-disambiguated PRED in practice. To test this hypothesis, we move the GCD-T partition layer to later model layers. If the models use ARG to contextualize PRED, then we should see PRED contributions increase in later layers.

Method. We compute the SPRL contributions but vary the partitioning layer (the point at which in-focus and out-of-focus elements are partitioned from one another)[10] over the 24 Transformer blocks of each model. The average percent-contribution is measured as a proportion of all event elements; that is, the percent-contributions of ARG, MOD, PRED, and OTHER sum to 100% in each layer.

Results. In the percent-contributions of RoBERTa (Figure 3a), the ARG and PRED embeddings contain approximately equal amounts of thematic information for the first 20 Transformer blocks. In the last few layers, however, the importance shifts towards ARG and MOD while the relative contributions of PRED and OTHER decrease. In GPT-2 (Figure 3b), the relative contribution of ARG gradually decreases as those of PRED and OTHER increase. As with RoBERTa, the direction of the trends switch after approximately layer 20.

These results suggest that GPT-2 may adhere to something like the hypothesized processing

[10] See §C.2 for more information about partitioning layers.

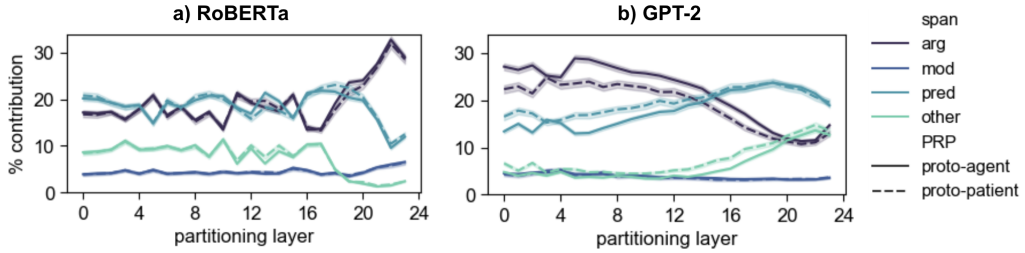


Figure 3: Average proportional GCD-T token contribution by partitioning layer. Ribbons represent 95% CI.

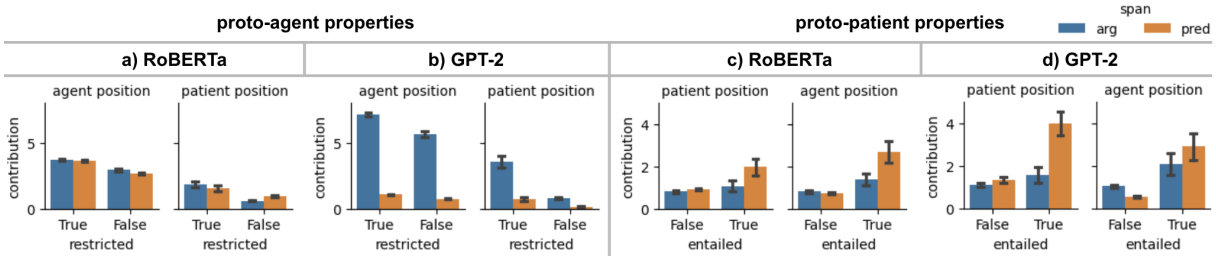


Figure 4: GCD-T SPRL positive token contributions by verbal lexical specification. Restricted denotes that the PRP is a selectional restriction and entailed denotes it is a lexical entailment. Agent-position denotes active subjects/passive objects; patient-position denotes active objects/passive subjects. Error bars represent 95% CI; note that errors are higher for proto-patient properties due to smaller sample sizes.

pipeline: initially focusing primarily on ARG, which is then used to reinterpret the semantic information from PRED. RoBERTa seems to do the opposite, contextualizing ARG using PRED. Future studies could explore which conditions give rise to different contextualization strategies, but this is beyond the scope of this paper.

5 Human Experiments

5.1 Research Questions and Design

Our GCD-T analysis revealed that LLMs rely on non-verb sentential elements to make PRP judgments, even for post-verbal arguments, and that this was especially true for proto-agentive properties. In this section we test whether this finding holds for humans as well. To that end, we elicited PRP judgments from humans using stimuli with various elements masked to see how masking of sentence components influenced their proto-agentive PRP inferences in pre-verbal and post-verbal contexts.

To elicit PRP judgments, we adopt the annotation paradigm of Reisinger et al. (2015) and Kako (2006). However, to isolate the contribution of each component (PRED, ARG, and MOD), we replace the other two regions with phonologically-plausible pseudo-words in the style of Berko (1958). In doing so, participants must use the semantic content of the remaining real word to make their PRP judgments. This allows us to assess the extent to which inferences are made on the basis of cues present in each element. Since our GCD-T analysis suggested that proto-agentive PRPs should be most influenced by non-verbal elements, we focused on the following proto-agentive PRPs highly correlated with animacy: `A`, `I`, `S`, and `V`.

Participants. We recruited 62 participants from the Prolific crowdsourcing platform,[11] all located in the United States and self-reported monolingual English speakers. Participants were compensated at a rate equivalent to $15/hour.

5.2 Experimental Setup

Materials. There were four levels of the real word factor (PRED, MOD, INANIMATE-ARG, and ANIMATE-ARG), which varied which of the potential PRP triggers was left as a real word while others were masked by pseudo-words. There was also a position factor with two levels, Subject and Object, which varied whether the argument of interest was the grammatical subject or object of an active transitive sentence allowing us to vary the target argument's syntactic position. These factors were fully crossed resulting in 8 total conditions. Stimulus sentences were all of the form "[Proper Name] [PRED]-ed the [MOD] [ARG]" or "The [MOD] [ARG] [PRED]-ed [Proper Name]".

We use state-of-mind adjectives for MOD as we hypothesize they are the most likely to increase animacy. However, all sentences include an adjective to control for any focus or salience effect result-

[11] https://www.prolific.com

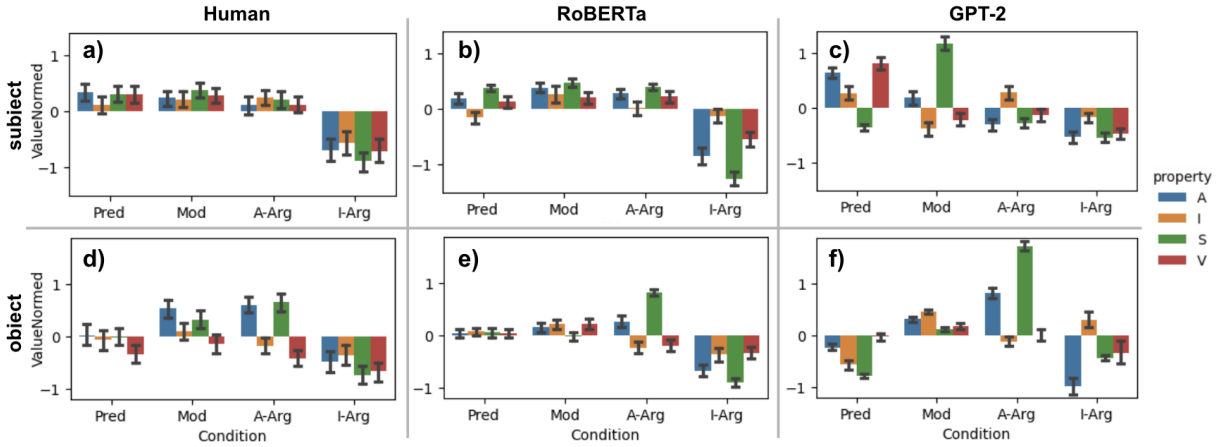


Figure 5: Average SPRL human ratings and model GCD-T contributions, z-scored by PRP and syntactic position. A-Arg denotes ANIMATE-ARG and I-Arg denotes INANIMATE-ARG. Error bars represent 95% CI.

ing from modifying an argument at all. Similarly, when the ARG is unmasked, we use two classes of nouns, Animate (animals) and Inanimate. The real PREDs were selected from a list of the most frequent active transitive verbs in Decomp V1. To study the influence of selectional restriction, we chose verbs which, according to VerbNet, require animate subjects, but may use both animate and inanimate objects.

We create 16 sets of stimuli, each with a unique real proper name, verb, adjective, inanimate noun, and animate noun. Example stimuli are shown in §E. Pseudo-words were generated using the Python package `gibberish`[12] and compared to transition probabilities for English orthography to ensure phonological plausibility. We also manually checked the generated pseudo-words to verify they do not look like misspellings of real English words nor can be pronounced as real English words. Proper names were gender-balanced.

Procedure. Prior to the experiment, participants underwent a training phase[13] and were told that they would be shown unfamiliar words and would need to use the context to figure out the meaning of each sentence. Participants saw only one of the eight sentence arrangements of each item, in addition to seeing only Subjects or only Objects, to reduce confusion and minimize potential confounding factors. They were asked to rate the likelihood of all four PRPs on each sentence however each sentence-PRP question pair was displayed on a different screen and randomized throughout the trials to reduce cross-contamination between PRP questions. As such, each participant saw 64 total experimental trials.

Models. In addition to collecting human judgments, we also obtained GCD-T contributions from RoBERTa and GPT-2 following the methods described in §4. To align with humans who give a single PRP rating, we sum all GCD-T logits, positive and negative, to get a single PRP rating.

To decrease the variance of the model results, we augmented the stimuli by generating all combinations of argument-predicate pairs, resulting in 256 sets. Unlike with humans, it is not problematic for the LLMs to see all arrangements of each set, as their parameters are frozen after fine-tuning on Decomp. As GCD-T allows for the contribution of each component in a sentence to be computed independently in a single forward pass, the use of pseudo-words is not necessary to probe the models. Thus, the models receive two sentences from each set, one with the ANIMATE-ARG and one with the INANIMATE-ARG, and both containing the PRED and MOD. PRED and MOD contributions are averaged across both sentences when the target argument is the object, but computed using only the ANIMATE-ARG with subject targets.[14]

5.3 Results and Discussion

Figure 5 shows the average contributions for the human annotators and LLMs, z-scored by syntactic position and PRP. RoBERTa accurately captures word-level human ratings in all conditions (Figure 5b, 5e), despite only being trained on

[12] https://github.com/greghaskins/gibberish

[13] The practice questions during the training phase were completely separate from the experiment and given only to familiarize participants with the 5-point Likert scale.

[14] This is because sentences with the INANIMATE-ARG subjects were ungrammatical due to violating the semantic restrictions of the PRED. Indeed, humans participants never see the true PRED and INANIMATE-ARG together, and likely assume the pseudo-word subject represents an animate noun.

annotations using full-sentence contexts. Of the tested properties, RoBERTa struggles most with I, however this property also has the lowest inter-annotator agreement in humans. Conversely, GPT-2 fails to predict human ratings, particularly in subjects (Figure 5c). This result makes sense since both RoBERTa and the humans have access to the full sentence context when making their judgments, while GPT-2 does not.

All analyses of the human data are done using the ANIMATE-ARG condition as the baseline for the comparison. We see that in the object syntactic position (Figure 5d), A and S judgments are significantly lower in PRED and INANIMATE-ARG compared to ANIMATE-ARG. The difference in S between MOD and ANIMATE-ARG is smaller but still significant. Additionally, the MOD condition shows moderately significant increases in V and I while INANIMATE-ARG showed a moderately significant decrease in V. These results show that MOD, ANIMATE-ARG, and INANIMATE-ARG have larger impacts on PRP inferences than PRED, which is notable given the centrality of predicates in conveying events. We will return to this in the General Discussion.

In the subject position (Figure 5a), S and V for the MOD condition and V and A for the PRED condition are significantly higher than the corresponding values for ANIMATE-ARG. These results suggest that, aside from S, active-voice subjects are agentive by default (as most lexical items in subject position don't influence PRPs) but can be manipulated by verbs and adjectives.

These human findings concur with our GCD-T results that non-verbal elements contribute to PRP inferences in readers. We even see, through the different conditions, that these various grammatical elements contribute to PRPs in distinct ways. Most notably, we see that MOD and ANIMATE-ARG increase A and S, especially in the object condition, while the INANIMATE-ARG condition decreases the inferences of all four properties.

6 General Discussion

We introduced a new method, Generalized Contextual Decomposition for Transformers (GCD-T), for feature attribution in LLMs, based on previous (Generalized) Contextual Decomposition methods for earlier neural architectures (Murdoch et al., 2018; Jumelet et al., 2019). GCD-T requires no additional training or gradient computation and is therefore less expensive than other probing methods. One common concern with LLM probing methods is that the probes used to interpret an LLM can produce false positives because the probes can learn to solve the probe tasks even when the LLM does not (Hewitt and Liang, 2019). GCD-T avoids this issue because it simply decomposes the representations within a model to disentangle the input-output mappings learned by the model during training.

We used GCD-T to investigate how different event elements contribute to the inference of semantic proto-role properties (PRPs) in Transformer language models. *A priori*, one could imagine several different ways that processors could infer PRPs. First, because events are communicated using predicates, one could imagine that processors would rely as much as possible on predicates, minimizing the ability of arguments to modulate PRP inferences. We find that is not the case, and that LLMs rely heavily on arguments and modifiers to infer PRPs. Second, readers could adopt a maximally incremental approach where they use pre-verbal arguments to infer PRPs but they neglect to use post-verbal arguments, having already seen the verb and much of the sentence. We show that they do not. Post-verbal material is used similarly to pre-verbal material. Finally, all PRPs could be inferred using the same procedure (e.g., all PRP inferences could rely 20% on the argument and 80% on the predicate). We find that this is not the case either: proto-agent properties rely more on argument cues while proto-patient properties rely more on predicate cues, though we find differences within these broad proto-role categories as well.

Finally, we validated the results of our computational experiments using a human experiment. We confirmed that humans also heavily use non-predicate elements to infer PRPs, even in post-verbal sentence contexts. In fact, we found that non-verbal elements had *larger* impacts on PRP inferences than predicates in our task. As with LLMs, each PRP was affected differently by our experimental manipulations, indicating that readers adopt different strategies for inferring each PRP. There remains much to do in this area, including analyzing more proto-role properties, and exploring the specific ways in which verbal selectional restriction and lexical entailments influence each PRP inference in humans, but we hope our work can serve as a first step in this direction.

References

Jean Berko. 1958. The child's learning of english morphology. *WORD*, 14:150–177.

Ina Bornkessel, Matthias Schlesewsky, and Angela D Friederici. 2003. Eliciting thematic reanalysis effects: The role of syntax-independent information during parsing. *Language and Cognitive Processes*, 18(3):269–298.

Wing-Yee Chow, Ellen Lau, Suiping Wang, and Colin Phillips. 2018. Wait a second! delayed impact of argument roles on on-line verb prediction. *Language, Cognition and Neuroscience*, 33(7):803–828.

Donald Davidson. 1967. Truth and meaning. In *Philosophy, Language, and Artificial Intelligence: Resources for Processing Natural Language*, pages 93–111. Springer.

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.

David Dowty. 1991. Thematic proto-roles and argument selection. *Language*, 67(3):547–619.

David R. Dowty. 1989. On the Semantic Content of the Notion of 'Thematic Role'. In Gennaro Chierchia, Barbara H. Partee, and Raymond Turner, editors, *Properties, Types and Meaning: Volume II: Semantic Issues*, pages 69–129. Springer Netherlands, Dordrecht.

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.

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.

E. Matthew Husband. 2023. Thematic separation in light of sentence comprehension. *Language and Linguistics Compass*, 17(4):e12496.

Ray Jackendoff. 1976. Toward an Explanatory Semantic Representation. *Linguistic Inquiry*, 7(1):89–150.

Jaap Jumelet, Willem Zuidema, and Dieuwke Hupkes. 2019. Analysing Neural Language Models: Contextual Decomposition Reveals Default Reasoning in Number and Gender Assignment.

Edward Kako. 2006. Thematic role properties of subjects and objects. *Cognition*, 101(1):1–42.

Angelika Kratzer. 1996. Severing the external argument from its verb. *Phrase structure and the lexicon/Kluwer*.

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.

Beth Levin. 2012. On Dowty's "Thematic proto-roles and argument selection". In *A Reader's Guide to Classic Papers in Formal Semantics: Volume 100 of Studies in Linguistics and Philosophy*, pages 103–119. Springer.

Chia-Hsuan Liao, Ellen Lau, and Wing-Yee Chow. 2022. Towards a processing model for argument-verb computations in online sentence comprehension. *Journal of Memory and Language*, 126:104350.

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

Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-collados. 2022. TimeLMs: Diachronic language models from Twitter. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 251–260, Dublin, Ireland. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

W. James Murdoch, Peter J. Liu, and Bin Yu. 2018. Beyond Word Importance: Contextual Decomposition to Extract Interactions from LSTMs. In *International Conference on Learning Representations*.

Terence Parsons. 1990. *Events in the Semantics of English: A study in subatomic semantics*. MIT Press.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Dee Ann Reisinger, Rachel Rudinger, Francis Ferraro, Craig Harman, Kyle Rawlins, and Benjamin Van Durme. 2015. Semantic Proto-Roles. *Transactions of the Association for Computational Linguistics*, 3:475–488.

Rachel Rudinger, Adam Teichert, Ryan Culkin, Sheng Zhang, and Benjamin Van Durme. 2018. Neural-Davidsonian Semantic Proto-role Labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 944–955, Brussels, Belgium. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Soumya Sanyal and Xiang Ren. 2021. Discretized integrated gradients for explaining language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10285–10299, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thibault Sellam, Steve Yadlowsky, Ian Tenney, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Raluca Turc, Jacob Eisenstein, Dipanjan Das, and Ellie Pavlick. 2022. The multiBERTs: BERT reproductions for robustness analysis. In *International Conference on Learning Representations*.

Lloyd S Shapley. 1953. A value for n-person games. *Contribution to the Theory of Games*, 2.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org.

Adam Teichert, Adam Poliak, Benjamin Van Durme, and Matthew Gormley. 2017. Semantic Proto-Role Labeling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Aaron Steven White, Dee Ann Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal Decompositional Semantics on Universal Dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, Austin, Texas. Association for Computational Linguistics.

Alexander Williams. 2015. *Arguments in Syntax and Semantics*. Cambridge University Press.

A Dataset Preprocessing And Examples

In this section, we describe in greater detail the datasets used for the SPRL task, Decomp V1 (Reisinger et al., 2015) and Decomp V2 (White et al., 2016), in addition to our preprocessing methods. Examples from the Decomp dataset for SPRL are shown in Table 3.

Decomp V1 contains PRP judgments of 9,738 predicate-argument pairs in the Penn Treebank WSJ corpus (Marcus et al., 1993). Annotators rated the likelihood of each pair and PRP on a 5-point Likert scale (1=*very unlikely*, 2=*somewhat unlikely*, 3=*not enough information*, 4=*somewhat likely*, and 5=*very likely*). Negated and modal verbs were removed.

Decomp V2 contains PRP judgments of 6,091 predicate-argument pairs from the Universal Dependencies English Web Treebank (Silveira et al., 2014). The same annotation protocol was used as V1, with some modifications, including switching from using one annotator to two for each pair.

In our combined dataset, we use the 10 PRPs annotated in both V1 and V2 (see Table 1). We use binned scores of > 3 and ≤ 3 following Teichert et al. (2017) and Rudinger et al. (2018). This corrects for class imbalances, as only 8.8% of pre-binarized scores are 2 or 4. As more than 96% of predicates are in active voice, we deterministically passivized active transitive verbs from the corpus to preserve semantic roles but balance syntactic positions. We manually checked augmented sentences to ensure grammaticality. We assume that passivization generally does not affect PRP ratings and so use the active PRP ratings for our generated passives; while passivization can result in decreased agentiveness (e.g., $5 \rightarrow 4$), we believe it is unlikely to cause a PRP rating to flip from ≤ 3 to > 3, or vice-versa. To offset the large number of active, non-transitive sentences, we upsampled the resulting passive sentences during training to ensure that the models saw a balanced number of passive and active structures. Dataset statistics are shown in Table 4.

We train on Decomp V1 and V2, but the results in Table 2 include only evaluation on Decomp V2 so that inter-annotator agreement can be used as a benchmark against which to compare LLM performance. Evaluation results on the Decomp V1 dataset in comparison to prior models are presented in §D, Table 6.

B Comparison of GCD and CD in Transformers

In this section, we compare the differences between the formulations of GCD (Jumelet et al., 2019) and CD (Murdoch et al., 2018) and explain why we chose to follow GCD for the decomposition of Transformers. Additionally, we provide empirical evidence using the subject-verb number agreement task that GCD provides more intuitive results when used to interpret the behaviour of Transformers than CD.

B.1 Theoretical Motivation for GCD

Any contextual decomposition method needs to decide how to allow the in-focus partition to interact with the out-of-focus partition. On the one hand, keeping the two partitions totally separate simplifies the interpretability objective. On the other hand, the interaction between the in-focus and out-of-focus elements could be critical to the model behavior at hand. The original Contextual Decomposition (CD) method (Murdoch et al., 2018) maximized the separation between the partitions by only considering representations in-focus when they solely consisted of in-focus interactions (rendering out-of-focus any representations that resulted from interactions with out-of-focus material). Formally, the hidden state at time t, $\mathbf{h}_\beta^{\ell,t}$, only consisted of information resulting from the interactions between $\mathbf{h}_\beta^{\ell,t-1}$ embeddings in the previous recurrent cell. In contrast, (Jumelet et al., 2019) argued that this tight in-focus constraint led to a loss of information, such as syntactic knowledge about *when* information should be expressed. Therefore, they allowed, for example, interactions between previous cell state values $\mathbf{c}_\beta^{\ell,t-1}$ and forget gate values $\mathbf{f}_\gamma^{\ell,t}$ to be included in current cell state $\mathbf{c}_\beta^{\ell,t}$, as $\mathbf{f}_\gamma^{\ell,t}$ is only used in the cell state to inform *which* of the $\mathbf{c}_\beta^{\ell,t-1}$ values should be passed on, and is not able to contribute other information.

Thus, following the reasoning of Jumelet et al., we chose not to decompose the attention weights in GCD-T to also allow for interactions between in-focus and out-of-focus queries and keys, so that both partitions may inform which of the in-focus values should be passed onto the next in-focus encodings. Otherwise, we keep the in-focus and out-of-focus partitions separate by decomposing the value vectors. However, we also confirm this decision experimentally.

Sentence	Predicate	Argument	I	V	A	S	EB	ED	EA	CoS	CoP	CoL
I e-mailed your assistant earlier this morning and have had no response .	e-mailed	I	5	5	5	5	5	5	5	1	1	3
		your assistant	1	1	2	5	5	5	5	1	1	1
In the meantime, we extend our best wishes for a safe and happy holiday season .	extend	we	5	5	5	5	5	5	5	1	1	3
		our best wishes for a safe and happy holiday season	1	1	1	1	3	5	5	1	1	1

Table 3: Examples of items in the Decomp dataset for the SPRL task. PRP judgements are shown on the 5-point Likert scale. PRP labels are, from left to right: INSTIGATION, VOLITION, AWARENESS, SENTIENT, EXISTEDBEFORE, EXISTEDDURING, EXISTEDAFTER, CHANGEOFSTATE, CHANGEOFPOSSESSION, and CHANGEOFLOCATION.

Dataset Split	Decomp V1				Decomp V2			
	Active		Passive		Active		Passive	
	Subject	Object	Subject	Object	Subject	Object	Subject	Object
Train	4097	3662	1520	1544	3898	1627	1864	1514
Validation	490	477	183	182	507	231	277	221
Test	477	488	200	177	517	249	263	241

Table 4: Distribution of annotated argument-predicate pairs in Decomp V1 and V2 dataset after passivized sentences were generated. Note that the models were trained and validated on both Decomp V1 and V2, but tested on only Decomp V2 for our analyses. The Object column includes both direct and indirect objects.

B.2 Comparison of GCD-T and CD-T on Number Agreement

We experimented with fully decomposing attention weights based on the method of CD (Murdoch et al., 2018), which we call Contextual Decomposition for Transformers (CD-T). CD-T attention weights are decomposed based on CD's (Murdoch et al., 2018) $f_t \odot c_{t-1}$ and $i_t \odot g_t$ decompositions, such that the β attention weights contain only information from the β and δ $\mathbf{Q}$ and $\mathbf{K}$.

We compare the results of CD-T to GCD-T on the Number Agreement task (see §C.1) in Figure 6. Results show that CD-T failed to accurately capture model behaviour. In BERT, CD-T incorrectly states that singular nouns provide evidence *against* predicting singular verbs, and in GPT-2, it states that plural nouns provide evidence against predicting plural verbs. GCD-T has the expected behaviour and shows that singular nouns provide evidence *for* predicting singular verbs and plural nouns provide evidence for predicting plural verbs in both models.

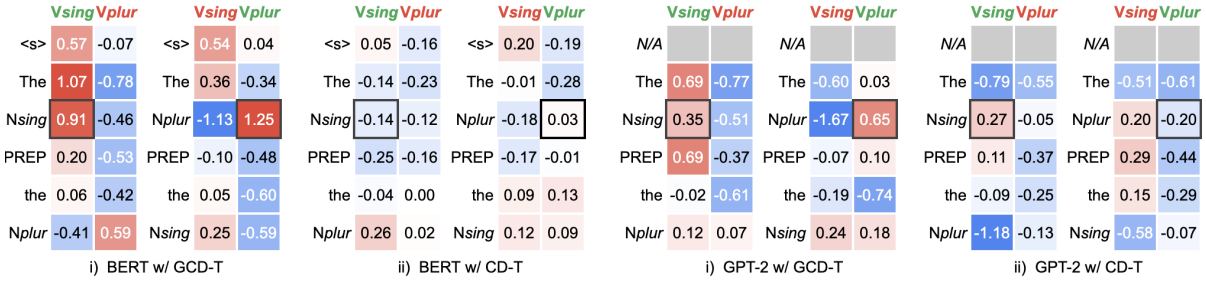
C Validating GCD-T

As GCD-T is a new method, prior to using it in our novel task of studying the locus of signal for proto-role properties, we choose to validate that the method works as intended on two other well-studied interpretability tasks: 1) subject-verb number agreement, which has been widely used in the interpretability literature and was used to motivate the original GCD work, and 2) sentiment analysis, which provides an indication of how the method works with richer semantic signals.

C.1 Number Agreement

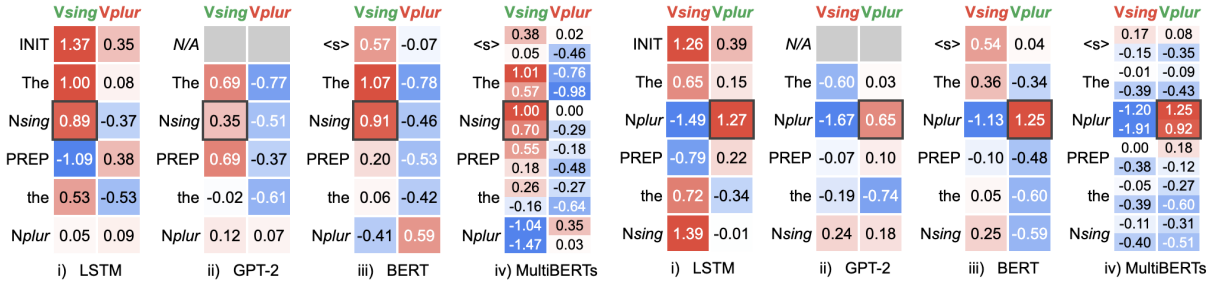
We validate GCD-T against GCD for LSTMs using a Number Agreement dataset (Lakretz et al., 2019) which tasks models with predicting the proper verb form when the sentence contains an intervening distractor of the opposite number. For example, given the sentence prefix "The boy near the cats ..." a model should assign higher probability to the singular continuation 'knows' than the plural 'know' due to the singular subject 'boy'.

We compute average token contributions of pretrained GPT-2-Medium (Radford et al., 2019) and BERT-base-uncased (Devlin et al., 2019) models on this dataset. We discard sentences with multi-token verbs to permit verb masking and we normalize contributions over all single-token verbs in



(a) BERT comparison (b) GPT-2 comparison

Figure 6: Average token contributions using GPT-T and CD-T for the number agreement task. The two columns in each group correspond to GCD-T/CD-T logits over singular and plural verb forms. Red cells indicate positive contributions of that row's token to the verb form tracked by that column (color scaled from [0,1]), while blue cells correspond to negative contributions (color scaled from [-1,0]).



(a) singular subject, plural attractor (b) plural subject, singular attractor

Figure 7: Average token contributions for the number agreement task. The two columns in each group correspond to GCD-T logits over singular and plural verb forms. Black borders indicate the subject's correct number. Red cells indicate positive contributions of that row's token to the verb form tracked by that column (color scaled from [0,1]), while blue cells correspond to negative contributions (color scaled from [-1,0]). INIT denotes initial LSTM state. PREP denotes preposition. MultiBERT values indicate upper and lower bound of the 95% CI across BERT 25 models.

the vocabulary,[15] using a complete verb list obtained from Google Books Ngram. To verify robustness, we also evaluate all MultiBERTs, a set of 25 BERT-base-uncased models pretrained from randomly initialized seeds (Sellam et al., 2022).

Results are presented in Figure 7 with the LSTM results of Gulordava et al. (2018) after reimplementing the GCD method of Jumelet et al. (2019).[16] As expected, GCD-T shows that in both BERT and GPT-2, singular nouns provide positive evidence in predicting singular verbs and plural nouns provide positive evidence for plural verbs. This pattern aligns with the LSTM contributions and demonstrates that the proposed GCD-T is comparable to the original GCD for LSTMs.

C.2 Sentiment Detection

GCD-T also provides insight into how word tokens become contextualized across the layers of the model, by permitting manipulation of the Transformer layer at which tokens are first partitioned into β and γ components. We call this the *partitioning layer*. By default, partitioning occurs prior to the first layer (layer 0), but withholding this until a higher layer allows for inter-component interactions to occur up to the partitioning layer. This application of GCD is an innovation of our work. To illustrate how contextualization analysis can be performed with GCD-T, we apply it to the widely used task of sentiment detection.

We additionally compare GCD-T to traditional Shapley values using the package `shap`.[17] The latter method computes model-level rather than

[15] Normalizing over verbs makes results more interpretable than prior methods since a value of 0 can then be interpreted as the token's logit contribution towards the average verb.

[16] Our LSTM numbers differ slightly from those reported in Jumelet et al. (2019) because we use our per-token contribution measure in our reimplementation of their code. We find the contribution measurements of Jumelet et al., $\mathbf{z}_\beta = \hat{\mathbf{z}}_\beta / \hat{\mathbf{z}} = \hat{\mathbf{z}}_\beta / (\hat{\mathbf{z}}_\beta + \hat{\mathbf{z}}_\gamma + \hat{\mathbf{z}}_\delta)$, are problematic when it is possible for the logits to be negative. For example, if $\hat{\mathbf{z}}$ and $\hat{\mathbf{z}}_\beta$ are both negative, then $\mathbf{z}_\beta > 0$, which is unintuitive given the in-focus partition contributed negatively. Furthermore, negative logit values can distort the denominator when summed with other, positive logit values.

[17] https://github.com/shap/shap

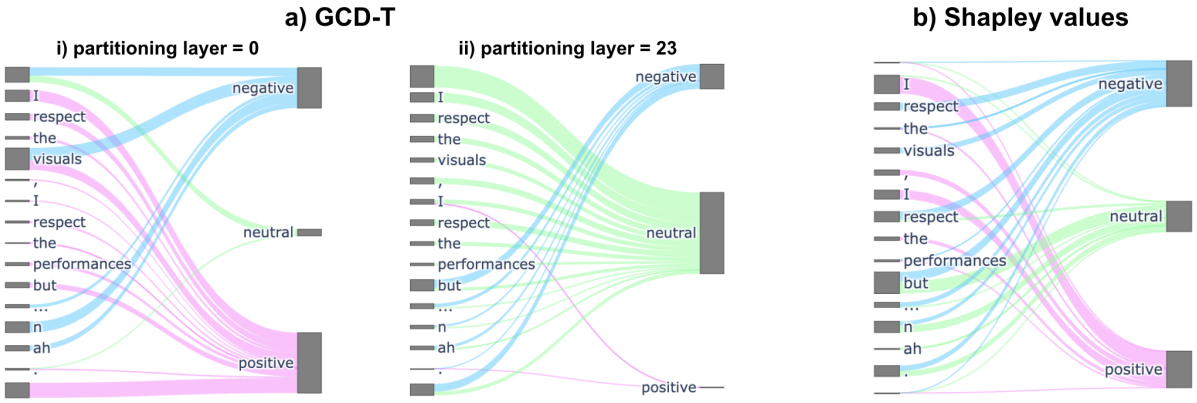


Figure 8: Example sentiment detection net-positive contributions to negative, neutral, and positive classes using a) GCD-T (layer-level decomposition) and b) Shapley values (model-level decomposition). The first and last token are the special tokens <*s*> and </*s*>.

layer-level decompositions, meaning `shap` gives less accurate estimations for large models like Transformers.

Figure 8 shows an example using a RoBERTa sentiment detection model (Loureiro et al., 2022) with a 1.5/5-star review for the movie *Joker: Folie à Deux* from the review aggregator website Rotten Tomatoes. With GCD-T, when components are partitioned before the first encoder layer (Figure 8a(i)), the start of the review ("I respect the visuals, I respect the performances") is interpreted as contributing a positive sentiment. When partitioning is done just prior to the final layer (Figure 8a(ii)), the words instead contribute neutrally to the sentiment. This is a result of the tokens getting contextualized across layers, likely because the review conclusion ("but... nah") forces a reinterpretation of the rest of the sentence. Conversely, the model-level Shapley value decomposition (Figure 8b) incorrectly shows "respect the visuals" and "respect performances" as contributing a negative sentiment.

Note that the model was originally trained to classify the sentiment of sequences, not individual tokens. Yet GCD-T still allows us to study the token level representational contributions learned by the model.[18] Because GCD-T does not require further model or probe training it is thus highly computationally efficient compared to other probing techniques. For instance, unlike Integrated Gradients (Sundararajan et al., 2017; Sanyal and Ren, 2021), GCD-T does not require the costly computation of gradients during the forward pass.

D SPRL Comparison to Prior Models

We compare the performance of RoBERTa and GPT-2 on the Decomp V1 dataset with models proposed by prior works: Logistic Regression (LR) (Reisinger et al., 2015), CRF (Teichert et al., 2017), and BiLSTM (Reisinger et al., 2015). Results are shown in Table 6. As we use the results reported by the original authors, the LR, CRF, and BiLSTM F1 scores are only for the sentences present in the original V1 dataset. Note that RoBERTa and GPT-2 models are additionally trained on Decomp V2 and passive sentences from our data augmentation process (described in §A), making the task more complex and the models more robust; however, there is a slight trade-off in performance on the original Decomp V1 dataset.

E Human Experiment Examples

For our human experiment, Table 5 shows an example set and Figure 9 shows the task given for the INANIMATE-ARG/Object arrangement.

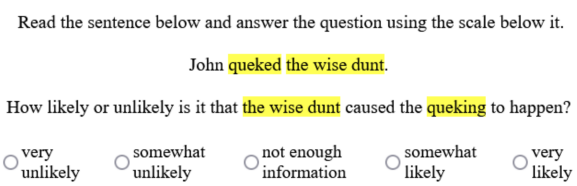


Figure 9: Example of display shown to participants.

[18] The original final classifier logits are equivalent to the summation of all GCD-T token contributions, plus the contribution of bias parameters.

Sentence	Condition	Position
John *wanted* the veect dunt.	PRED	Object
John queked the *wise* dunt.	MOD	
John queked the veect *bird*.	ANIMATE-ARG	
John queked the veect *book*.	INANIMATE-ARG	
The veect dunt *wanted* John.	PRED	Subject
The *wise* dunt queked John.	MOD	
The veect *bird* queked John.	ANIMATE-ARG	
The veect *book* queked John.	INANIMATE-ARG	

Table 5: Example set of stimulus in all arrangements. Emphasis not shown to participants.

	LR	CRF	BiLSTM	RoBERTa		GPT-2	
				Active	Passive	Active	Passive
instigation	76.7	85.6	88.6	84.4	90.6	82.8	92.2
volition	69.8	86.4	88.1	86.5	93.1	86.2	92.5
awareness	68.8	87.3	89.9	89.8	94.2	87.8	94.0
sentient	42.0	85.6	90.6	84.1	93.5	84.5	93.9
existed_before	79.5	84.8	85.1	86.9	89.4	85.9	86.8
existed_during	93.1	95.1	95.0	93.5	96.9	94.0	96.7
existed_after	82.3	87.5	85.9	87.1	90.3	85.9	89.0
Δ_state	54.6	66.1	71.0	67.5	69.6	67.0	70.1
Δ_possession	0.0	38.8	58.0	61.8	71.8	61.8	71.8
Δ_location	6.6	35.6	45.7	53.4	51.3	53.2	58.5

Table 6: Binary F1 scores of models on Decomp V1.

Quantifying mutual intelligibility gradients in Turkic languages using language models

Moldir Baidildinova, Shiva Upadhye, Austin Wagner, Connor Mayer, Richard Futrell
{mbaidild,upadhyes,wagnera3,cjmayer,rfutrell}@uci.edu
Department of Language Science
University of California, Irvine

1 Introduction

Mutual intelligibility (MI) refers to the ability of speakers of one language to understand another language, often due to genealogical relatedness that results in lexical and grammatical correspondences. MI is a gradient phenomenon because it exists on a continuum from near-total comprehension to no understanding, often showing asymmetry between related language pairs (Chambers and Trudgill, 1998; Tang and van Heuven, 2007; van Bezooijen and Gooskens, 2005). A prime example is the Turkic language family, which is often described as highly similar from a historical-typological perspective, with substantial overlap across multiple linguistic dimensions, which is commonly assumed to support MI within the family (Kornfilt, 2018; Tambovtsev, 2001; Baskakov, 1988; Lindsay, 2010). These languages share substantial lexical similarity, characterized by a high percentage of shared cognates, as well as grammatical structures, word order and phonetic-phonological overlap, as shown in Table 1. Also, languages within the same branch tend to show higher levels of MI, partly because of their close proximity within the genealogical tree, or what has been discussed in terms of cophenetic distance (Gooskens et al., 2018). In other words, the closer two languages are on the tree, the higher their intelligibility rates tend to be. For instance, Turkish and Azerbaijani are generally more mutually intelligible (Salehi and Neysani, 2017) than Turkish and Kazakh or Turkish and Uyghur because they belong to the same Oghuz branch; similar patterns can be observed across other language pairs as well.[1] Taken together, these patterns suggest that MI is best understood as a multidimensional phenomenon, shaped by the combined effects of lexical, grammatical, and phonetic-phonological similarity.

Language	Branch	IPA transcription
Turkish	Oghuz	(ben biɾ) kitap okujoɾum.
Azerbaijani		(mæn) kitɑb oxujuɾɑm.
Kazakh	Kipchak	(men) kɪtɑp oqəp ʒɑtərmən.
Kyrgyz		(men) kitep okup ʒɑtɑm.
Uzbek	Karluk	(men) kitɔb oqijapman.
Uyghur		(mæn) kitɑb oquwɑtimæn.

Table 1: MI among Turkic languages illustrated by the sentence: *I am reading a book.*

However, existing computational approaches typically quantify MI using lexical or phonetic distances computed over hand-curated cognate sets or high-frequency word lists, and most of such work has focused on Indo-European languages (Gooskens and van Heuven, 2021; Nieder and List, 2024). Here, we propose a neural language modeling-based approach to investigating MI within the Turkic language family. Rather than relying on hand-curated features, our approach learns these regularities directly from naturalistic text. We then ask to what extent the resulting similarity estimates track MI among Turkic languages.

2 Current study

We hypothesize that greater overlap across multiple linguistic dimensions facilitates cross-lingual transfer and contributes to higher MI between languages within the same Turkic subgroup. Under this view, a model trained on one Turkic language should generalize more successfully to closely related target languages than to distantly related ones, thereby reflecting the gradient nature of MI. To test this hypothesis, we fine-tune these pre-trained models on target languages that vary in their degree of

[1] To the best of our knowledge, systematic human data on MI patterns within the Turkic family remain limited. Existing observations are often anecdotal, based on reports from Turkologists, native speakers or L2 learners who note, for example, that knowledge of Uyghur may facilitate understanding of Uzbek more than Kazakh.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 442–446
July 3-4, 2026 ©2026 Association for Computational Linguistics

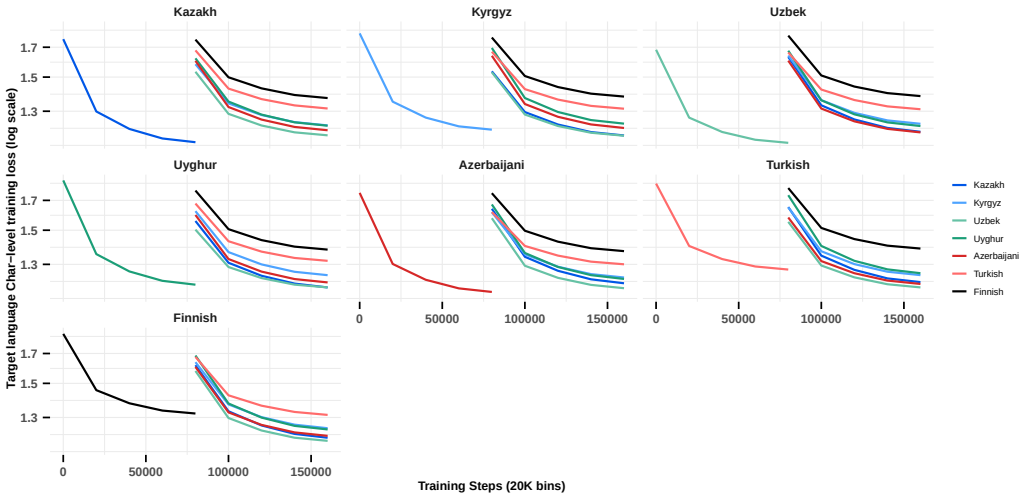


Figure 1: Character-level CE loss across training steps; models trained on source languages, then fine-tuned on target languages.

genealogical relatedness. We operationalize generalizability using character-level *cross-entropy (CE) loss*, which quantifies how surprised the model is, on average, by the phoneme sequences it encounters. Lower training and test CE loss are therefore expected to reflect stronger cross-lingual transfer, which we interpret as evidence of greater linguistic overlap between closely related language pairs. This provides a more fine-grained measure of cross-lingual transfer than curated cognate lists, which are limited to lexical overlap. In addition, we expect closer language pairs to show better overall learning dynamics across training, measured by *the Area Under the Curve (AUC)*, where smaller AUC values indicate better learning. We also expect more distant language pairs to show higher *rates of change (ROC)* during the early stages of training, due to larger gaps in lexicon and sound patterns. Taken together, our findings provide preliminary evidence that CE loss, AUC and ROC can approximate MI patterns across six Turkic languages.

3 Method

3.1 Training Datasets

We extracted raw text for closely related Turkish-Azerbaijani, Kazakh-Kyrgyz, and Uzbek-Uyghur language pairs from the OSCAR[2] corpus, filtered with FastText language identification (≥ 0.95 confidence) to reduce contamination (Abadji et al., 2022; Joulin et al., 2016). To approximate pronunciation across Turkic languages and avoid script inconsistency issues within the language family (Mirzakhalov et al., 2021), we transcribed all texts into broad IPA using established language-specific rule sets[3] (Zimmer and Orgun, 1992; Mokari and Werner, 2017; McCollum and Chen, 2021; McCollum, 2020; Ido, 2025; Mayer, 2021; McCollum, 2021). For each language, the transliteration output was verified to match exactly the IPA rules established in prior phonological analyses.

3.2 LSTM model

To mitigate tokenizer design issues in highly agglutinative, low-resource languages (Toraman et al., 2023), we trained a 2-layer character-level LSTM with 128-dimensional embeddings and 256 hidden units (Hochreiter and Schmidhuber, 1997). The model training followed a sequential bilingual training regimen (Arnett et al., 2025): a model was first trained and validated on a source language, then fine-tuned on a target language (Arnett et al., 2025) for each of the six languages. We also include Finnish as a distractor language, since it is typologically similar (agglutinative and vowel-harmonic) but genealogically unrelated (Suomi et al., 2009) to Turkic languages. In addition, we also manipulated exposure order (e.g., Kazakh-Kyrgyz, Kyrgyz-Kazakh), which yielded a total of 42 models. All models were exposed to 1 million tokens per language, split 75-25 into training and validation sets.

3.3 What is driving model convergence?

To shed light on which factors track the patterns observed in CE loss, AUC, and ROC, we examined four predictors of cross-lingual similarity: *cophenetic distance, lexical similarity, weighted*

[2]Open Super-large Crawled Aggregated coRpus

[3]The IPA rules are available on GitHub. The transliteration suite is available on Hugging Face.

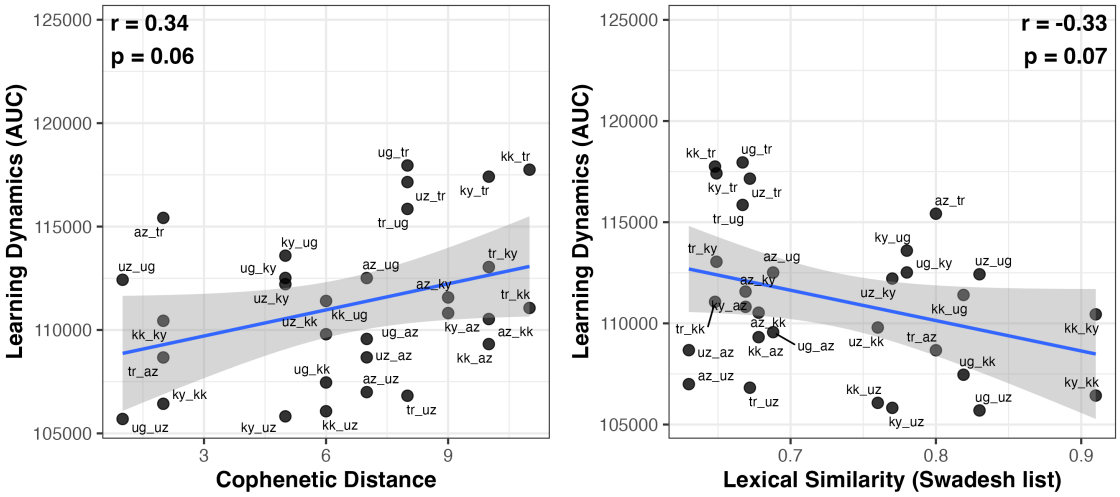


Figure 2: The AUC plotted against the cophenetic distance and lexical similarity.

trigram frequency overlap, and *vowel harmony index*. Cophenetic distance was computed following Gooskens et al. (2018) as the number of nodes separating two languages in the genealogical tree via their lowest shared ancestor node; the Turkic language tree was based on the reconstruction by Savelyev and Robbeets (2020). Lexical similarity was derived from the Swadesh-based estimates in Lindsay (2010). Weighted trigram frequency overlap was computed as a proxy for phonotactic similarity by extracting IPA-based character trigrams from size-normalized corpora, retaining frequent trigrams, and comparing their frequency counts across language pairs using weighted Jaccard similarity (Niwattanakul et al., 2013). We also computed a vowel harmony index for each training dataset following Harrison et al. (2004) and derived difference in the vowel harmony index between source and target languages. Finally, we correlated CE loss, AUC, and ROC with these predictors to evaluate which dimensions of linguistic similarity best account for model convergence across Turkic language pairs.

4 Results

Across experiments, CE loss broadly aligned with MI gradients: closer Turkic pairs showed smaller loss spikes, faster convergence, and lower training loss. Transfer was better from Kyrgyz to Kazakh than to Turkish, from Turkish to Azerbaijani than to Kazakh, and from Uyghur to Uzbek than to Kazakh, as shown in Figure 1. Although these differences were modest, they preserved the expected MI ordering, suggesting that CE loss is sensitive to different degree of overlap across Turkic pairs, with lower CE loss observed for closely related pairs within the same branch, such as Kazakh–Kyrgyz within the Kipchak branch, Turkish–Azerbaijani within the Oghuz branch, and Uzbek–Uyghur within the Karluk branch. Most importantly, the models appear to exhibit sensitivity to directional asymmetries between language pairs. For instance, Kazakh seems to converge to Kyrgyz more slowly than Kyrgyz converges to Kazakh, with similar patterns observed for Azerbaijani $<$ Turkish and Uzbek $<$ Uyghur. In addition, Turkish appears to be the least advantageous target language for most source models, whereas Uzbek shows the opposite pattern, serving as the most advantageous target language across all five source models. Finnish yielded the highest test loss in all transfer directions, consistent with its genealogical distance from the Turkic languages. We also correlated CE loss with cophenetic distance and lexical similarity. The correlations were positive but non-significant, tentatively suggesting that more cophonetically distant pairs may incur higher loss. Differences in the vowel harmony index and trigram frequency overlap did not yield systematic patterns.

As predicted, the AUC pattern was qualitatively lower for closer language pairs and for pairs with greater lexical overlap, as shown in Figure 2. Asymmetric patterns between language pairs are also evident in this plot, and these patterns are the same as those discussed above: Kyrgyz $<$ Kazakh, Azerbaijani $<$ Turkish, and Uzbek $<$ Uyghur. Dif-

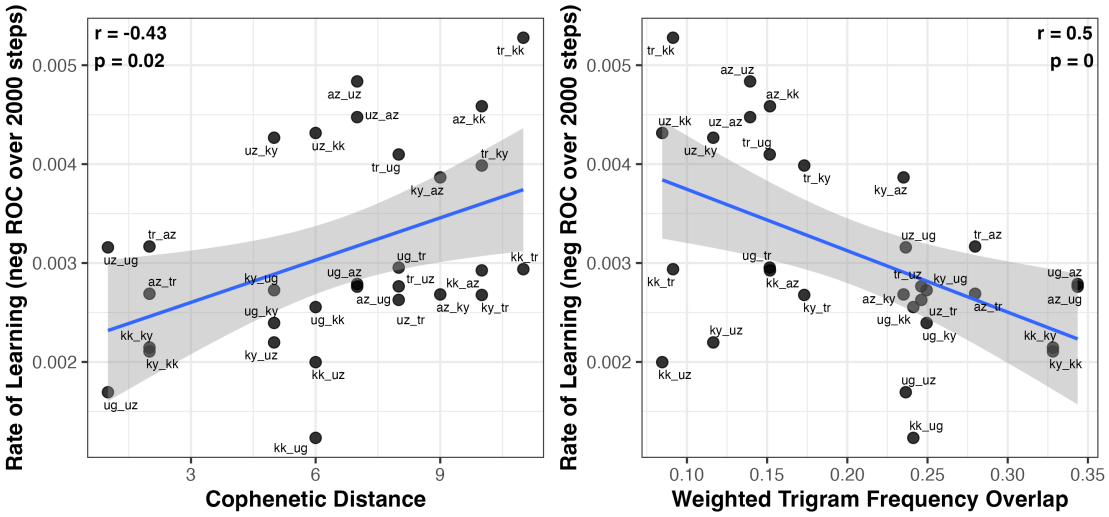


Figure 3: The ROC over 2000 steps of training plotted against cophenetic distance between language pairs and the weighted trigram frequency overlap.

ferences in the vowel harmony index did not show a consistent relationship with AUC across language pairs; a similar pattern was observed for trigram frequency overlap as well. In contrast, greater language distance appears to predict greater learning change: more cophenetically distant pairs show higher ROC, whereas pairs with greater trigram overlap show lower ROC, as shown in Figure 3. Unlike CE loss and AUC, ROC captures early-stage adaptation during training, suggesting that the rate of change may provide an additional indicator of model performance.

5 Discussion

Our results suggest that character-level LSTM models trained on naturalistic text can broadly capture MI patterns within the Turkic family. Across experiments, more closely related pairs generally showed lower CE loss and smaller AUC, consistent with greater cross-lingual generalizability. This suggests that even a relatively simple and scalable modeling approach can track MI gradients from phoneme-level distributional patterns in naturalistic text. The results also suggest that multiple dimensions of linguistic similarity may contribute to model-based MI patterns. We tested several possible predictors of model convergence. Overall, lexical similarity, trigram overlap, and cophenetic distance appeared to be the most informative predictors: models showed better transfer when languages shared more lexical and local phonotactic structure and when they were genealogically closer. In contrast, differences in vowel harmony index did not show a systematic relationship with the model metrics, suggesting that variation in the degree of vowel harmony does not have a strong effect on model performance in this setup.

The Finnish results further support the importance of genealogical relatedness. Although Finnish shares some typological properties with Turkic languages, such as agglutinative morphology and vowel harmony, it consistently produced the highest test loss across transfer directions. The models also appear to be sensitive to directional asymmetries between language pairs, suggesting that transfer is not simply determined by overall pairwise similarity. Instead, some source languages provide more advantageous transfer than others, even within related pairs. Of the measures we used to assess learning, ROC showed the clearest systematic pattern, with more distant pairs showing higher rates of change during the early stages of training, while pairs with greater trigram overlap showed lower ROC. This suggests that ROC may capture early-stage adaptation when the model encounters less familiar phoneme-sequence patterns.

Taken together, these findings suggest that character-level CE loss, AUC, and ROC can approximate MI patterns across Turkic languages, especially when similarity is grounded in lexical overlap, local phonotactic overlap, and historical relatedness. Ongoing collection of native-speaker MI judgments will allow us to test whether human speakers perceive these similarity and asymmetry patterns in the same way as the models.

References

Julien Abadji, Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2022. Towards a cleaner document-oriented multilingual crawled corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC 2022)*, Marseille, France. European Language Resources Association.

Catherine Arnett, Tyler A Chang, James A Michaelov, and Benjamin K Bergen. 2025. On the acquisition of shared grammatical representations in bilingual language models. *arXiv preprint arXiv:2503.03962*.

N. A. Baskakov. 1988. *Istoriko-tipologicheskaya fonologiya tyurkskikh yazykov*. Nauka, Moscow.

J.K. Chambers and Peter Trudgill. 1998. *Dialectology*. Cambridge University Press.

Charlotte Gooskens and Vincent J. van Heuven. 2021. Mutual intelligibility. In Marcos Zampieri and Preslav Nakov, editors, *Similar Languages, Varieties, and Dialects: A Computational Perspective*, Studies in Natural Language Processing, pages 51–95. Cambridge University Press, Cambridge.

Charlotte Gooskens, Vincent J Van Heuven, Jelena Golubović, Anja Schüppert, Femke Swarte, and Stefanie Voigt. 2018. Mutual intelligibility between closely related languages in Europe. *International Journal of Multilingualism*, 15(2):169–193.

David Harrison, Emily Thomforde, and Michael O'Keefe. 2004. The vowel harmony calculator. *Online: http://www. swarthmore. edu/SocSci/harmony/public_html*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Shinji Ido. 2025. Uzbek. *Journal of the International Phonetic Association*, 55(1-2):152–168.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Jaklin Kornfilt. 2018. Turkish and the Turkic languages. In *The world's major languages*, pages 536–561. Routledge.

Robert Lindsay. 2010. Mutual intelligibility among the Turkic languages. *Beyond Highbrow*.

Connor Mayer. 2021. *Issues in Uyghur backness harmony: Corpus, experimental, and computational studies*. University of California, Los Angeles.

Adam G McCollum. 2020. Vowel harmony and positional variation in Kyrgyz. *Laboratory Phonology*, 11(1).

Adam G McCollum. 2021. Transparency, locality, and contrast in Uyghur backness harmony. *Laboratory Phonology*, 12(1).

Adam G McCollum and Si Chen. 2021. Kazakh. *Journal of the International Phonetic Association*, 51(2):276–298.

Jamshidbek Mirzakhalov, Anoop Babu, Duygu Ataman, Sherzod Kariev, Francis Tyers, Otabek Abduraufov, Mammad Hajili, Sardana Ivanova, Abror Khaytbaev, Antonio Laverghetta Jr, and 1 others. 2021. A large-scale study of machine translation in the Turkic languages. *arXiv preprint arXiv:2109.04593*.

Payam Ghaffarvand Mokari and Stefan Werner. 2017. Azerbaijani. *Journal of the International Phonetic Association*, 47(2):207–212.

Jessica Nieder and Johann-Mattis List. 2024. A computational model for the assessment of mutual intelligibility among closely related languages. In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 37–43.

Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu. 2013. Using of Jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists*, volume 1, pages 380–384.

Mohammad Salehi and Aydin Neysani. 2017. Receptive intelligibility of Turkish to Iranian-Azerbaijani speakers. *Cogent Education*, 4(1):1326653.

Alexander Savelyev and Martine Robbeets. 2020. Bayesian phylolinguistics infers the internal structure and the time-depth of the Turkic language family. *Journal of Language Evolution*, 5(1):39–53.

Kari Suomi, Juhani Toivanen, and Riikka Ylitalo. 2009. *Finnish sound structure: Phonetics, phonology, phonotactics and prosody*. University of Oulu.

Yuri Tambovtsev. 2001. The phonological distances between Turkic languages based on some phonological features of consonants. *Asian and African Studies*, 10(1):11–43.

Chia-Jung Tang and Vincent J. van Heuven. 2007. The role of typological and lexical similarity in the perception of Chinese dialects. *Journal of Chinese Linguistics*, 35(2):309–326.

Cagri Toraman, Eyup Halit Yilmaz, Furkan Şahinuç, and Oguzhan Ozcelik. 2023. Impact of tokenization on language models: An analysis for Turkish. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(4):1–21.

Renée van Bezooijen and Charlotte Gooskens. 2005. How easy is it for speakers of Dutch to understand Frisian and Afrikaans, and why? *Lingua*, 115(10):1479–1500.

Karl Zimmer and Orhan Orgun. 1992. Turkish. *Journal of the International Phonetic Association*, 22(1-2):43–45.

Algebraic Classification of Reduplicative Processes

Matthew Hayden
Stony Brook University
matthew.hayden@stonybrook.edu

Abstract

This paper offers an updated perspective on the computational complexity of reduplication. Since one-way deterministic transducers cannot model reduplication in a straightforward way, the phenomenon has long been considered the outlier of morphology from a complexity perspective. Drawing on algebraic methods, I show that the vast majority of reduplicative processes belong to a few remarkably simple classes of subregular functions. A detailed study of the RedTyp database (Dolatian and Heinz, 2019) reveals that 100% of the surveyed reduplicative processes correspond to string-to-string functions in the class $\mathbb{D}\mathbf{A}$, while over 98% are locally testable ($\mathbb{L}\mathbf{J}_1$) and over 87% are locally trivial ($\mathbb{L}\mathbf{1}$). These results indicate a new upper bound on the complexity of reduplication that is comparable to that of morphological processes in general.

1 Introduction

The starting point for much work in computational morphology is to regard morphological processes as string-to-string functions. Morphological investigation from the perspective of *subregular linguistics* tries to identify the simplest subclasses of string-to-string functions that account for the range of attested processes. It is well-known that one-way deterministic finite-state transducers (1DFTs) are sufficient to capture almost all attested morphological transformations, with total reduplication ($w \rightarrow ww$) being the exception (Roark and Sproat, 2007). In fact, the vast majority of phonological and morphological transformations can be modeled with input strictly local (ISL) or output strictly local (OSL) functions, which are restricted subclasses of the functions computable by a 1DFT (Chandlee, 2014). Again, Chandlee (2014) notes that total reduplication is the exception. Given that most everything else in morphology appears to be subregular, it seems odd that total reduplication is uniquely challenging to compute, despite occuring in around 85% of languages in the World Atlas of Language Structures (WALS) (Rubino, 2013). This apparent imbalance of complexity has remained a puzzle in subregular morphology. Prior research has explored what changes can be made to the conventional finite-state machinery in order to correctly model reduplication (Dolatian and Heinz, 2020, cf. Wang, 2021). This paper pursues an alternative approach that uses tools from the algebraic theory of semigroups to identify universal properties of attested reduplicative functions.

Reduplication is a typologically pervasive phenomenon wherein some amount of phonetic material is copied to induce a change in meaning. Consider the following examples, from Indonesian (Cohn, 1989) and Samoan (Moravcsik, 1978) respectively:

(1) buku → bukubuku
'Book' → 'Books'

(2) alofa → alolofa
'He loves' → 'They love'

While total reduplication, as in (1), is most common, 75% of languages in WALS additionally have partial reduplication, as in (2). In partial (or bounded) reduplication, a subcomponent of the input known as the *target* is copied to create the *reduplicant*. For instance, the target in (2) is the syllable *lo* in *alofa*, while the reduplicant is the second *lo* in *alolofa*. In cases of total (full, or unbounded) reduplication, the target of the reduplication is the entire input form. While the bounded nature of partial reduplication renders it computable by 1DFT (Chandlee, 2014), one-way transducers are incapable of computing total reduplication ($w \rightarrow ww$), because they make only one pass over the input.

More recently, Dolatian and Heinz (2020) argue that both total *and* partial reduplication are better represented by two-way deterministic finite-state transducers (2DFTs) which are strictly more

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 447–459
July 3-4, 2026 ©2026 Association for Computational Linguistics

powerful than their one-way counterparts. These machines are able to compute functions in the class of concatenated OSL functions (C-OSL), which intuitively correspond to the transformations computable by multiple OSL functions with concatenated outputs. The class of C-OSL functions is shown to be sufficiently expressive for the majority of attested reduplicative patterns. Thus, C-OSL offers a tighter characterization of the space of morphological processes than does the full range of functions computable by 2DFTs (i.e. the *regular* functions).

However, Lambert and Heinz (2024) recently leveraged algebraic methods to reclassify phonological processes typically regarded as OSL (or related output-oriented classes) with respect to the properties satisified by their *syntactic semigroups*. Given a 1DFT that models a concrete phonological transformation, one can mechanically construct its syntactic semigroup, which serves as an algebraic description of the effect different alphabet symbols have on the states of the machine. The resulting semigroup structure can be probed with the algebraic decision procedures documented by Lambert (2022). Lambert and Heinz (2024) used these techniques to show that those OSL transductions actually have semigroups in relatively weak subregular classes. This paper extends that line of investigation to the two-way transducers largely classified as C-OSL by Dolatian and Heinz (2020).

The first contribution of this paper is an algorithmic explanation of how to construct transition semigroups for two-way machines, toward the end of facilitating future research in this vein. While the knowledge of how to do this has been implicit in the literature at least as far back as Birget (1989), I provide explicit instructions for how to infer the basis of the transition semigroup of an arbitrary two-way machine. The second contribution is a direct extension of the efforts of Dolatian and Heinz (2020) in the classification of reduplication, by way of a comprehensive study of the RedTyp database (Dolatian and Heinz, 2019). The third contribution comes in the form of a Python script for computing two-way semigroups automatically, and for the classification of the transducers in RedTyp.

Section 2 gives an overview of and motivation for two-way models of reduplication (Dolatian and Heinz, 2019, 2020). In Section 3, I show how to classify reduplicative processes with respect to the algebraic properties satisfied by their finite-state implementations. In Section 4, I apply these methods to the finite-state transducers catalogued in RedTyp, a database of reduplicative patterns across 91 languages (Dolatian and Heinz, 2019). Ultimately, the results suggest that reduplication is not as computationally complex as it is thought to be—the majority of reduplicative processes fall into three or more rather simple subclasses of the algebraic subregular hierarchy.

2 Background

One of the coarsest classifications of a transformation is whether it can be modeled by a one-way deterministic finite-state transducer (1DFT), a machine that computes a left-to-right (or right-to-left) function from an input string to an output string. The symbols in the input come from a finite alphabet Σ, and the symbols in the output string come from a finite alphabet Ω. In this paper, all the transducers happen to have identical input and output alphabets. I use the abbreviation $\Sigma_{\rtimes}$ when Σ is enriched with two delimiters $\rtimes$ and $\ltimes$ that mark the left and right edges of a word.

Definition 2.1 (1DFT)**.** A one-way deterministic finite-state transducer is a four-tuple $\langle Q, q_i, F, \Delta \rangle$, where Q is a set of states, $q_i \in Q$ is the initial state, $F \subseteq Q$ is the set of final states, and Δ is a set of transition instructions. The instructions in Δ are of the form $(p, i, q, o) \in Q \times \Sigma_{\rtimes} \times Q \times \Omega^*$ such that Δ contains at most one instruction corresponding to each $(p, i) \in Q \times \Sigma_{\rtimes}$.

While it is well-known that 1DFTs cannot compute total reduplication, Dolatian and Heinz (2020) argue that 1DFT models also fail to capture partial reduplication in a linguistically plausible way. This argument appeals to the *origin semantics* (Bojańczyk, 2014), which formalize the correspondence between the input and output symbols in a transduction.

As an example, consider the leftmost diagram Figure 1, which describes the one-way computation of $\rtimes$CVC$\ltimes$ $\mapsto$ CVCVC. The lines indicate the correspondences between the input and the output. A one-way machine that computes this function starts on the left edge symbol $\rtimes$, then moves left-to-right through the string, writing symbols to the output. Once the machine sees the first V, it outputs VCV, so as to emulate initial-CV reduplication. This diverges from linguistic intuitions because the C in the reduplicant is associated with a V in the base. In this respect, one-way models fall short of correctly capturing the dependencies linguists

Figure 1: Origin semantics of Initial CV Reduplication from a 1DFT (left) and a 2DFT (right)

are trying to model. Therefore, they argue that reduplication necessitates the jump to two-way deterministic finite-state transducers (2DFTs), which can compute a more complex class of functions. 2DFTs achieve this greater expressive power by removing the restriction that input strings be processed strictly left-to-right or right-to-left, so that the machine may move back and forth on the input during the course of computation.

Definition 2.2 (2DFT)**.** A two-way deterministic finite-state transducer is a four-tuple $\langle Q, q_i, F, \Delta \rangle$, where Q is a set of states, $q_i \in Q$ is the initial state, $F \subseteq Q$ is the set of final states, and Δ is a set of transition instructions. The instructions in Δ are of the form $(p, i, q, o, d) \in (Q \times \Sigma_{\rtimes} \times Q \times \Omega^* \times D)$ such that Δ contains at most one instruction corresponding to each $(p, i) \in Q \times \Sigma_{\rtimes}$, and where where $D = \{1, -1\}$. When an instruction has $d = 1$, computation proceeds by moving one symbol to the *right* in the input string. When $d = -1$, computation proceeds by moving one symbol to the *left* on the input string.

The diagram on the right in Figure 1 demonstrates a two-way transducer's origin semantics on the same computation. Whereas the one-way machine is forced to move in one direction, the two-way machine is allowed to revisit the first C in the input before writing the reduplicant. Thus, both the Cs and Vs in the reduplicant are associated correctly with their correspondents in the base. So, with respect to the origin semantics, 2DFTs are better suited for computing partial reduplication than 1DFTs. Given that 2DFTs are also capable of computing total reduplication, it seems that 2DFTs are a better model for reduplicative processes in general.

In fact, Dolatian and Heinz (2020) show that virtually all reduplicative processes are modelable with 2DFTs, and offer a computational typology of reduplication by identifying weaker subclasses of 2DFTs that maintain strong empirical coverage. Notably, approximately 87% of RedTyp transducers were shown to be C-OSL functions. However, while OSL functions (and related classes) have often been invoked in the analysis of non-local phonological transformations, algebraic analysis reveals that these seemingly output-oriented functions fall into more constrained classes when classified only with respect to their inputs. Lambert and Heinz (2024) take a second look at which are phonological processes in the literature that originally motivated OSL, namely iterative spreading (Chandlee et al., 2015), and (a)symmetric harmony (Heinz, 2010). While these long-distance processes were previously understood as OSL (or tier-based OSL) patterns, their semigroup structures reveal them to be in the subregular algebraic class $\mathbb{L}\mathbf{1}$, or its tier-based extension. Known also as the *locally trivial* or *generalized definite* class, $\mathbb{L}\mathbf{1}$ constraints also naturally accomodate local phonotactics (Lambert, 2026), and so membership in $\mathbb{L}\mathbf{1}$ is taken to be a reasonable level of complexity for a phonological process or pattern.

That these OSL functions have such restricted subregular semigroups is of particular interest because, as proven by Lambert (2022), the class OSL resists an algebraic characterization. It turns out that *any* finite semigroup is the syntactic semigroup of some OSL transducer, meaning the examined output-oriented processes in principle could have had algebraic structures of arbitrary regular complexity. Given that the current understanding of reduplication relies on C-OSL, it stands to wonder whether reduplicative transducers also have surprisingly simple semigroups, potentially shifting our estimation of reduplicative complexity in line with the rest of morphophonology. To test this hypothesis, I apply algebraic methods to a database of reduplicative transducers, generating and classifying semigroups for each with a Python script, ultimately yielding a robust typology of reduplicative complexity.

3 Methods

When it comes to reduplication, 2DFT models seem prefable to 1DFT models, both because they can compute total reduplication and because they yield the appropriate origin semantics in cases of partial reduplication. While reduplicative 2DFTs are known to be mostly C-OSL (Dolatian and

Heinz, 2020), the algebraic approach may uncover a computationally simpler generalization. This section describes how to investigate this hypothesis, with concrete results from a database study. Section 3.1 describes the database itself, Section 3.2 details how to construct semigroups for two-way transducers, and Section 3.3 provides classification procedures for the subregular classes which dissect the space of attested reduplications.

3.1 Database: RedTyp

RedTyp (Dolatian and Heinz, 2019) is a typological database of reduplication that comes furnished with finite-state models of 138 reduplicative processes. These processes span 91 languages, ultimately consisting of 57 distinct 2DFTs. The data in RedTyp was drawn largely from surveys (Moravcsik, 1978; Rubino, 2005; Inkelas and Downing, 2015) with the goal of collecting diverse processes. While fairly comprehensive, it should be noted that RedTyp obviously does not contain an exhaustive list of all documented reduplicative transformations. Even still, its empirical coverage and supply of finite-state models make RedTyp a database conducive to computational study.

Although RedTyp contains 138 examples of reduplication according to Dolatian and Heinz (2019), I was only able to extract the data of 136 processes via SQL query. Three of those processes were missing the field corresponding to the 2DFT, leaving 133 transducers available for study. A text file containing all 133 2DFTs, as well as the code for computing and classifying their semigroups, is available on Github.[1]

3.2 Semigroups for two-way transducers

Each example of reduplication catalogued in RedTyp is associated with a 2DFT that can be classified algebraically in a procedural way. The current work on algebraic classification (see: Lambert, 2026 and references therein) extracts syntactic semigroups from one-way machines, which can then be mechanically checked for set membership across many subregular classes. However, there are a notions of semigroups for 2DFTs as well. I adopt the definition of *transition semigroup*[2] for two-way automata mentioned originally by (Birget, 1989) and explicated more recently by (Carton and Dartois, 2015; Kunc and Okhotin, 2011). The extraction of the transition semigroup from a 2DFT is also mechanical, and the subregular decision procedures apply to all semigroups in the same way, regardless of whether they were obtained from one-way or two-way machines. Therefore, it is possible to automate the classification of RedTyp transducers to get an informative view of the reduplicative complexity landscape in one fell swoop. In this section, I detail how to construct these transition semigroups step-by-step.

As a running example of semigroup construction, consider this pattern of initial consonant reduplication, found in Agta (Moravcsik, 1978):

(3) liw → lliw
'scold' → 'angry'

The first step in the classification of a morphological processes is to settle on an analysis and a 2DFT. The transducer given in Figure 2, which models this process in Agta, is adapted only slightly from its instantiation in RedTyp (for details, see the discussion in Limitations).

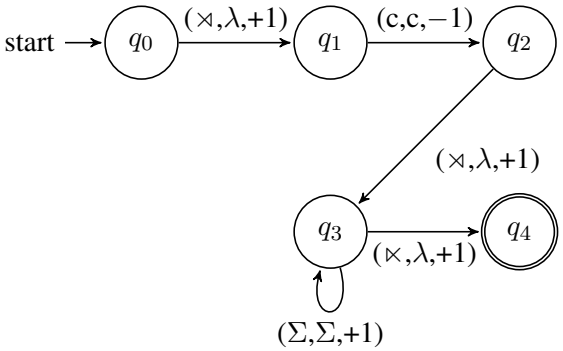


Figure 2: 2DFT for Agta Initial-C Reduplication

When computing the transition semigroup for a one-way machine, one partitions Σ^+ into the equivalence classes of input words that have the same effect on the states of the machine. The question is: Starting from a state p, when an input string is processed left-to-right (or right-to-left), in which state q does the machine end up? In a two-way machine, however, there are more ways that an input string can be processed besides just left-to-right or right-to-left. Because the machine can move back and forth on the input, it is possible to start on the left edge of the input string and exit on the left edge. Likewise, it is possible to start on the right edge and exit the string on the left. This leaves four possibilities: a string can be processed left-to-right,

[1] www.github.com/mphayden1/classifying-reduplication

[2] The syntactic semigroup of a 1DFT corresponds to the transition semigroup of its *canonical* form; for 2DFTs, no canonical form is known (see Limitations).

left-to-left, right-to-right, and right-to-left. These are called the *behaviors* of a string.

Definition 3.1 (Behaviors)**.** Let $\mathcal{A} = \{Q, q_i, F, \Delta\}$ be a two-way automaton, and let $w \in \Sigma^+$. Then the left-to-left behaviors of w are given by:
$b_{ll}(w) = \{(p, q)$: There exists a run over $\mathcal{A}$ such that $\mathcal{A}$ starts on the leftmost symbol of w in state p and leaves w from the left edge in state q.$\}$
The other three behaviors, $b_{lr}, b_{rr},$ and b_{rl} are defined analogously. Let $B(w) = \langle b_{ll}, b_{lr}, b_{rl}, b_{rr} \rangle$.

Notice that for a given single symbol $a \in \Sigma$, it is necessarily the case that $b_{ll}(a) = b_{rl}(a)$, and that $b_{lr}(a) = b_{rr}(a)$. This is because the leftmost and rightmost symbol are the same for a unigram.

Definition 3.2 (Transition Semigroup)**.** A semigroup is a set that is closed under an associative binary operation. Let $\mathcal{A} = \{Q, q_i, F, \Delta\}$ be a two-way automaton. Let $\sim_B$ be the equivalence relation such that for $u, v \in \Sigma^+$, $u \sim_B v$ iff $B(u) = B(v)$. Then the *transition semigroup* of $\mathcal{A}$ is given by the quotient $\Sigma^+ / \sim_B$.

For a two-way machine $\mathcal{A}$, despite a possibly infinite input language consisting of words $w \in \Sigma^+$, there is a finite set of behaviors $S = \{B(w_1), B(w_2), \ldots, B(w_k)\}$ such that the behavior of every word in Σ^+ has a behavior in S. That set, when equipped with a particular product operation $\bullet$, forms the transition semigroup $\langle S, \bullet \rangle$. The $\bullet$ operation was originally defined by Birget (1989), and it describes how to faithfully compose the behaviors $B(u), B(v)$ of two words in order to recover the behavior of their concatentation $B(uv)$.

Theorem 3.1 (Concatenation of inputs)**.** Let $\mathcal{A} = \{Q, q_i, F, \Delta\}$ be a two-way automaton. Let u and v be words over Σ^+, with behaviors $B(u)$ and $B(v)$. Let $\bullet$ be the operation defined as:

$$
\begin{aligned}
&B(u) \bullet B(v) = \\
&\langle b_{ll}(u) \cup b_{lr}(u)[b_{ll}(v)b_{rr}(u)]^* b_{ll}(v) b_{rl}(u), \\
&\ b_{lr}(u)[b_{ll}(v)b_{rr}(u)]^* b_{lr}(v), \\
&\ b_{rl}(v)[b_{rr}(u)b_{ll}(v)]^* b_{rl}(u), \\
&\ b_{rr}(v) \cup b_{rl}(v)[b_{rr}(u)b_{ll}(v)]^* b_{rr}(u) b_{lr}(v) \rangle
\end{aligned}
$$

The juxtaposition of behaviors $b_{xy}(w)b_{x'y'}(w')$ denotes their composition as relations. The star * is the reflexive-transitive closure of the relation enclosed in brackets. Then, the following formula holds:

$$B(u) \bullet B(v) = B(uv)$$

The key fact is that the behaviors of a word uv can be inferred directly from the behaviors of u and v, which entails that the entire transition semigroup is generated from the behaviors of the single-letter words in Σ. These could be thought of as the primitive behaviors, which form the basis of the semigroup. Once the primitive behaviors are known, one can discover the rest of the semigroup elements through their repeated $\bullet$-composition, until no new behaviors are discovered.

The second step, then, is to identify these primitive behaviors. To identify the primitive behaviors of a two-way machine, consider one symbol at a time, and consider its effect on the machine starting at each state. Refer back to the Agta transducer $\mathcal{A}$ in Figure 2. This 2DFT operates over alphabet symbols c and v, which means that we only need to directly calculate $B(c)$ and $B(v)$, and the rest of the semigroup elements will follow. Recalling that $b_{ll}(c) = b_{rl}(c)$ and $b_{lr}(c) = b_{rr}(c)$, it suffices to check only the left-left and left-right behaviors of c. In Figure 2, the only states that have outgoing arrows labeled with c are q_1 and q_3. Because the c-transition from q_1 to q_2 has a -1 as its direction, it moves the head of the transducer to the left of the C in the input. Thus, $\mathcal{A}$ leaves starts in q_1 on the left edge of C, and leaves c on the left edge in state q_2. Since this is the only place in the machine where this action occurs, we conclude that $b_{ll}(c) = b_{rl}(c) = (q_1, q_2)$. The only other c-transition in $\mathcal{A}$ occurs as the self-loop on q_3. It has a positive direction component, meaning $\mathcal{A}$ starts on the left edge of c in state q_3 and exits c on the right in state q_3, so $b_{lr}(c) = b_{rr}(c) = (q_3, q_3)$. Performing the same procedure for v yields the following two behaviors, for $B(c)$ and $B(v)$:

$$\langle \{(q_1, q_2)\}, \{(q_3, q_3)\}, \{(q_1, q_2)\}, \{(q_3, q_3)\} \rangle$$
$$\langle \emptyset, \{(q_1, q_2), (q_3, q_3)\}, \emptyset, \{(q_1, q_2), (q_3, q_3)\} \rangle$$

The third step is to $\bullet$-compose each of the primitive behaviors with each other, and then with the products of those compositions, and so on, until no new behaviors are found. Following the formulas given in 3.1, we can compute $B(cv)$ directly from the behaviors of c and v. This yields a distinct behavior:

$$B(cv) = \langle \{(q_1, q_2)\}, \{(q_3, q_3)\}, \emptyset, \{(q_3, q_3)\} \rangle$$

While writing in-line, I use the notation [w] to refer to the semigroup element corresponding to $B(w)$. Because $B(c)$, $B(v)$, and $B(cv)$ are all

distinct, that yields at least three semigroup elements: [c], [v], and their product [cv]. In fact, these three elements are all and only those in $S = \Sigma^+/\sim_B$. One can check this by verifying [v] • [c] = [v] • [v] = [v], and so on until every pairwise combination has been tried. Every •-composition of two of these elements in S evaluates to another in S, so S is closed under •-composition. The structure of the semigroup $\langle S, \bullet \rangle$ is summarized in Table 1.

•	c	v	cv
c	cv	cv	cv
v	v	v	v
cv	cv	cv	cv

Table 1: Multiplication table for Agta

To some extent, these can be reasoned about with common sense. In S, [v] corresponds to all words starting with a vowel, which are rejected. So, concatenating anything on the end of a word with behavior [v] should evaluate to another word with behavior [v], since the new word would still start with a vowel. The only string with behavior [c] is c itself, and any word that starts with c and is longer than one symbol has behavior [cv]. Thus, concatenating anything to the end of these words will yield a word with behavior [cv].

3.3 Classifying two-way semigroups

The final step is the classification of the semigroup, in accordance with the decision procedures collected by Lambert (2022). I detail only some of these procedures here, which suffice to test membership in the subregular classes that turn out to be relevant for reduplication. Figure 3 provides an inclusion chart for the algebraic classes at hand, where **1** corresponds to a point in the intersection of **D** and **K**.

As discussed previously, much work has gone into finding subclassifications of 1DFTs that provide a tight fit for the types of phonological processes that are attested. For example, ISL and OSL functions, and their tier-based extensions (Chandlee et al., 2014). A function f is k-ISL if there is some finite k such that the output of the function at any step in the computation is decided by the last $k-1$ symbols in the *input*. We say f is ISL if it is k-ISL for any k. Analogously, f is k-OSL if the output at any point in computation is decided by the last $k-1$ symbols in the *output*. Together, these classes offer impressive empirical coverage (for details, see Chandlee, 2014; §7.1). While the ISL class covers local phonological processes, the ability for OSL functions to track output symbols allows them to model long-distance processes, like iterative spreading.

Coming from another angle, the algebraic subregular hierarchy presented in (Lambert, 2022) provides a fine-grained taxonomy of string-to-string functions, plus a suite of algorithmic decision procedures for the classification of transducers. These two perspectives are connected: Lambert and Heinz (2023) showed that the class of total ISL functions is exactly the algebraic class **D** of the *definite* functions. On the other hand, however, the OSL class has no algebraic correspondent. In fact, for any finite semigroup S, there exists an OSL function which has S as a semigroup (Lambert, 2022). This motivated Lambert and Heinz (2024) to reexamine many prototypical examples of (tier-based) OSL functions from through the lens of algebra. Each of the processes, when classified according to semigroups, were shown to belong to the classes **D**, **K**, $\mathbb{L}$**1**, or their tier-based extensions. These classes are near the absolute bottom of the subregular hierarchy—only slightly above the class **1** of *trivial* semigroups. My investigation of the semigroup structure of reduplicative 2DFTs reveals that reduplication likewise has a rather simple computational core.

3.3.1 Idempotence and algebraic identities

The property of idempotence is an important notion in the algebraic study of semigroups, and it plays a critical role in probing subregular complexity. Let S be some semigroup, and $a \in S$. Then a is called *idempotent* if and only $aa = a$. An *aperiodic* semigroup has the special property that all elements a eventually yield an idempotent under repeated composition, meaning there is some k such that $a^n = a^{n+1}$ for all $n \geq k$. For a given a in an aperiodic semigroup, we write a^ω to refer to the unique idempotent obtained under the repeated squaring of a. Then a semigroup S is aperiodic if and only if the identity $a^\omega = a^{\omega+1}$ holds for all $a \in S$. A famous result by Schützenberger (1965) holds that the languages with aperiodic syntactic monoids are precisely those belonging to the subregular class of star-free.

It turns out that every reduplicative transducer in RedTyp is in fact aperiodic. For example, by inspecting Table 1, it is clear that [v] and [cv] are idempotent elements. And while [c] is not an

idempotent, it has an idempotent power, because $[c]^3 = [c] \bullet [c] \bullet [c] = [cv] \bullet [c] = [cv]$. However, this is not especially surprising; after all, star-free is a rather robust class, subsuming most of the subregular classes of linguistic interest. However, there are more restricted subregular classes that correspond to more constraining algebraic identities. The following theorem collects identities from Lambert (2022) which pick out the relevant classes for reduplication. These classes are relatively weak, indicating that reduplicative processes actually cluster near the bottom of the algebraic subregular hierarchy.

Theorem 3.2. The following hold:

- **1** A semigroup S is *trivial* iff it satisfies $x = y$ for all $x, y \in S$.
- **D** A semigroup S is *definite* iff $xa^\omega = a^\omega$ for all $x, a \in S$.
- **K** A semigroup S is *reverse definite* iff $a^\omega x = a^\omega$ for all $x, a \in S$.
- $\mathbb{L}$**1** A semigroup S is *generalized definite* iff $a^\omega x a^\omega = a^\omega$ for all $a, x \in S$.
- $\mathbb{L}\mathbf{J}_1$ A semigroup S is *locally testable* iff $a^\omega x a^\omega y a^\omega = a^\omega y a^\omega x a^\omega$ for all $a, x, y \in S$.
- $\mathbb{D}\mathbf{A}$ A semigroup S is in $\mathbb{D}\mathbf{A}$ iff $(xyz)^\omega y (xyz)^\omega = (xyz)^\omega$ holds for all $x, y \in S$.

Determining which of these identities hold in a semigroup is a matter of straightforward symbol manipulation. For Agta, notice that $[v] \bullet [c] = [v] \bullet [v] = [v] \bullet [cv] = [v]$, and $[cv] \bullet [c] = [cv] \bullet [v] = [cv] \bullet [cv] = [cv]$. This process therefore satisfies the identity $a^\omega x = a^\omega$ for all $x, a \in S$, and so belongs to the class **K** of *reverse definite functions*. By inclusion, this process is also locally trivial, locally testable, and $\mathbb{D}\mathbf{A}$. These facts can be verified by testing the other identities in Theorem 3.2. In the appendix, results for the RedTyp transducers across these six classes are collected in tables.

4 Results

The algebraic analysis of the transducers in RedTyp reveals three subregular classes within which the overwhelming majority of reduplication is contained. Of these classes, the least restricted is $\mathbb{D}\mathbf{A}$, which is the smallest class that contains all 133 transducers. Out of those 133 transducers, 131 were classified as $\mathbb{L}\mathbf{J}_1$, and so $\approx 98\%$ of RedTyp is contained in the intersection of $\mathbb{D}\mathbf{A}$ and $\mathbb{L}\mathbf{J}_1$, which are incomparable (as shown in Figure 3). 116 of the

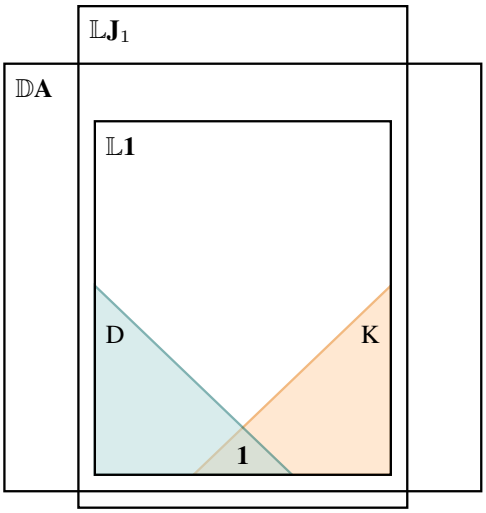


Figure 3: Algebraic subregular hierarchy for reduplication

transducers fall into $\mathbb{L}\mathbf{1}$, which is contained in $\mathbb{L}\mathbf{J}_1$, and so $\approx 87\%$ are also in $\mathbb{D}\mathbf{A} \cap \mathbb{L}\mathbf{1}$. All cases of unbounded reduplication are trivial (**1**), which is the simplest possible type of string-to-string function from the algebraic perspective (a fact first noted by Lambert, 2022).

The classes with the broadest coverage—$\mathbb{D}\mathbf{A}$, $\mathbb{L}\mathbf{J}_1$, $\mathbb{L}\mathbf{1}$—are all of theoretical interest. The class $\mathbb{D}\mathbf{A}$ is exactly the class $\mathrm{FO}^2_<$—that is, the class of functions definable in the fragment of first order logic restricted to two variables, and with the precedence relation $<$ (Thérien and Wilke, 1998). Moreover, these are the class of string functions defined by "partially-ordered" 2DFTs (Schwentick et al., 2001). Intuitively, a partially-ordered 2DFT is a two-way transducer satisfying the constraint that all loops in the machine are self-loops.

Regarding $\mathbb{L}\mathbf{J}_1$, their membership in this class establishes a connection with the locally testable languages (McNaughton, 1974), which are definable in the fragment of propositional logic over substrings (Rogers and Lambert, 2019). Since we can think of $\mathbb{L}\mathbf{J}_1$ transducers as a straightforward generalization of $\mathbb{L}\mathbf{J}_1$ acceptors, local testability of a *function* implies that there is some number k such that the output of the function at any step in computation is decided by the set of seen k-substrings and the last $k-1$ symbols seen. Recalling that local processes are generally computable by ISL_k functions, in which the output is decided only by the last $k-1$ symbols, it seems that the extra power needed to reduplication can be attained by also tracking the seen k-substrings in memory.

The generalized definite class, $\mathbb{L}\mathbf{1}$, is a class of mounting importance in computational mor-

phophonology. $\mathbb{L}\mathbf{1}$, perhaps with tiers, seems to capture a wide range of phonotactic patterns while being sufficienty constrained to be learnable (Lambert, 2026). Again, these results are concerned with sets of patterns, not functions, but an $\mathbb{L}\mathbf{1}$ acceptor becomes an $\mathbb{L}\mathbf{1}$ transducer when outputs are added to the transitions, and those two objects are treated the same under the algebraic approach. Lambert and Heinz (2024) use this to argue that the motivating cases for OSL are better thought of as as (tier-based) $\mathbb{L}\mathbf{1}$. This paper demonstrates the same for over 87% of reduplicative transducers, which indicates that, in broad strokes, reduplication is computationally aligned with morphophonology as a whole.

Discussion

The results are easier to interpret when taking a look at reduplicative processes that distinguish between the different classes. The weakest class in the subregular algebraic hiearchy is $\mathbf{1}$, which is the class of functions with trivial semigroups consisting of exactly one idempotent element. Intuitively, this means that the machine treats every string over the alphabet in the same way—it makes no distinction between different segments at any point in the computation. There are 40 processes in RedTyp that belong to this extremely restricted class, and they are are precisely the transducers for total reduplication and triplication. Moving up in complexity, a vast majority of processes have semigroups in $\mathbb{L}\mathbf{1}$, which establishes a computational parallel between reduplication and phonology in general. In fact, while Lambert (2026) argues that the tier-based extension of $\mathbb{L}\mathbf{1}$ is a tight approximation of phonotactic complexity, the reduplicative transducers in $\mathbb{L}\mathbf{1}$ need not reference tiers at all. More interesting, then, are the functions which are at least locally testable ($\mathbb{L}\mathbf{J}_1$). An example comes from Chamorro, which has a reduplicative process that targets the CV in the stressed syllable (Inkelas and Downing, 2015).

(4) hu.gán.do → hu.gá.gan.do
'play' → 'playing'

It makes sense that this process should require more expressive power than those in $\mathbb{L}\mathbf{1}$. $\mathbb{L}\mathbf{1}$ string functions are sensitive to the beginning and end of the input string, but the process in (4) requires a sensitivity to primary stress, which could be arbitrarily deep in the input word.

Out of 133 processes studied, only two had semigroups outside of $\mathbb{L}\mathbf{J}_1$. One of these is heavy CVC reduplication in Ilocano, which can mark plurality (Raimy, 2000):

(5) a. kaldíŋ → kal-kaldíŋ
'goat' → 'goats'

b. paʔíd → paa-paʔíd
'fan' → 'fans'

In (5a), the onset and vowel of the first syllable are copied, along with the next consonant. However, if the proceeding consonant is a glottal stop, as in (5b), the vowel is lengthened instead. Why this subtlety corresponds to a measurable increase in complexity is not clear. The corresponding transducer in RedTyp is complex, with 14 states, and it is possible that an alternative analysis or implementation may reduce its apparent complexity. The other process outside of $\mathbb{L}\mathbf{J}_1$ comes from Malay, and appears to be an instance of back-copying, where the reduplicant undergoes a phonological change that is also reflected in the base. Under the analysis of McCarthy and Prince (1995), in (6) the nasality spreads progressively from the nasal across the reduplicant, and then transfers back to the first vowel by correspondence.

(6) /hamə/ → [hãmə̃ hãmə̃], (*[hamə̃ hãmə̃])
'germ' → 'germs'

This particular example is also discussed by Dolatian and Heinz (2020), who also identify this process as unusually complex compared to others. They show that (6) cannot be computed by a C-OSL function, or even by the concatenation of two unrestricted (not necessarily OSL) 1DFTs. While there may be alternative simplifying analyses or two-way machines that have weaker semigroups, I leave this puzzle for future investigation.

The results of this study suggest that $\mathbb{D}\mathbf{A}$ is a hard upper bound on the complexity of reduplication (though likely not the least upper bound). As mentioned above, functions in $\mathbb{D}\mathbf{A}$ correspond to transformations definable in $\mathrm{FO}^2_<$—so what logically possible reduplications are predicted to be unattested? Let $\mathrm{FO}_<$ be the class of all first-order definable string transductions under the general precedence relation $<$. This class, defined in logic, actually coincides with the class of aperiodic transductions (Carton and Dartois, 2015). $\mathrm{FO}^2_<$ is contained in $\mathrm{FO}_<$, so it follows that reduplication is aperiodic, a fact we corroborate algebraically in Section 3.3.1. Under this view, we do not expect to find periodic (non-$\mathrm{FO}_<$) reduplication. This would

be like having a rule that says, "if *w* is of even length, output *ww*; otherwise, reduplicate only the first CV."

However, the bound of $\text{FO}^2_<$ also rules out the $\text{FO}_<$ functions that require more than two variables to define. One dividing line between $\text{FO}^2_<$ and $\text{FO}_<$ is that the idea of *betweenness* is inexpressible without access to a third variable (Krebs et al., 2020). This means that it is impossible to define regular properties like "an *a* can only precede another *a* if there is a *b* between them" in $\text{FO}^2_<$. We should not expect, then, to see a generalization like "total reduplication, but only if the stem has a vowel between every consonant." Whether or not that prediction bears out universally (though the results suggest it may)—this serves as a nice example of how the algebra makes predictions that are formally precise and eminently testable.

Future directions

One goal of this paper is to attest that algebraic classification is flexible, and can be a fruitful perspective, so long as the linguistic representation at hand is amenable to an algebraic analysis. While I only discussed one- and two-way machines in this paper, there are many equivalent representations for regular string functions that could also be explored. For instance, streaming string transducers are an equivalent model to two-way transducers, that also have a well-defined transition semigroup (Carton and Dartois, 2015). Or, one could forgo the machine entirely, and adopt a representation that uses *characteristic functions* instead (Bojańczyk, 2014), or one that visualizes transductions as languages of *origin graphs* (Bojańczyk et al., 2017). The theory of characteristic functions in particular could potentially be used to strengthen the results in this paper, as a characteristic function serves as the canonical device for a transduction with origin information. Looking beyound string-based representations, future research in this direction could consider trees and tree algebras, which could reveal algebraic insights in the domain of syntax or prosodic stucture.

5 Conclusion

This paper intersects two recent lines of inquiry in computational morphophonology: the development of linguistically plausible finite state models of reduplicative processes (Dolatian and Heinz, 2020) and the classification of phonological transformations using algebraic techniques (Lambert, 2022). I demonstrated how to calculate the syntactic semigroup of a 2DFT, provided a script for doing so automatically, and completed a study of RedTyp, which samples reduplicative processes from 91 languages. Ultimately, it yielded a typology of reduplicative functions and their distribution over some of the simplest subregular classes. That the vast majority of said functions ended up in $\mathbb{D}\mathbf{A} \cap \mathbb{L}\mathbf{J}_1$ identifies a new lowest upper bound on the complexity of reduplication. While the results of the algebraic classification are formal, they are of empirical interest because they describe constraints on the space of possible reduplicative processes that were previously unknown.

Importantly, the results of this study contrast with the traditional perspective on the computational complexity of reduplication. Because one-way machines fail to compute reduplication, it seems as if there is a large jump in computational power needed. In reality, this is not the case—the ability to move back and forth on the input is necessary to capture the correct origin semantics, but reduplicative processes seem not to make use of the extra power it affords. A key observation is that 1DFTs must compute *order-preserving* transductions (Bojańczyk, 2014), which cannot generate crossing associations in the origin semantics (as in Figure 1). This seems to be the property that sets reduplicative maps apart, and motivates the switch to a two-way machine. So, setting order-preservation aside, reduplication seems computationally unexceptional with respect to other morphological processes, spanning classes from $\mathbf{1}$ to $\mathbb{L}\mathbf{J}_1$ and $\mathbb{D}\mathbf{A}$. This upper bound is comparable with that of other complex phenomena outside of reduplication, e.g. ATR vowel harmony in Tutrugbu (Lambert, 2022; §5.6). Given also that RedTyp is majority generalized definite ($\mathbb{L}\mathbf{1}$), reduplication arguably seems run-of-the-mill in computational morphology and phonology. Not only does the algebraic perspective shed new light on the nature of reduplication, then, but also proves its worth as a sophisticated diagnostic for complexity that avoids conflating subregularity with order-preservation.

Acknowledgements

I am grateful to Jeffrey Heinz, Thomas Graf, and Mark Aronoff for their feedback and encouragement. I also thank the three anonymous reviewers for their helpful and detailed suggestions, as well

as the audience of the 2025 Rutgers Subregular Phonology Workshop for entertaining an early presentation of this work.

References

Jean-Camille Birget. 1989. Concatenation of inputs in a two-way automaton. *Theoretical Computer Science*, 63(2):141–156.

Mikołaj Bojańczyk. 2014. Transducers with origin information. In *Automata, Languages, and Programming: 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II 41*, pages 26–37. Springer.

Mikołaj Bojańczyk, Laure Daviaud, Bruno Guillon, and Vincent Penelle. 2017. Which classes of origin graphs are generated by transducers? In *ICALP 2017*.

Olivier Carton and Luc Dartois. 2015. Aperiodic two-way transducers and fo-transductions. In *24th EACSL Annual Conference on Computer Science Logic*, volume 41, pages 160–174.

Jane Chandlee. 2014. *Strictly local phonological processes*. Ph.D. thesis, University of Delaware.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2014. Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics*, 2:491–504.

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.

Abigail C. Cohn. 1989. Stress in indonesian and bracketing paradoxes. *Natural Language & Linguistic Theory*, 7(2):167–216.

Hossep Dolatian and Jeffrey Heinz. 2019. RedTyp: A database of reduplication with computational models. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 8–18.

Hossep Dolatian and Jeffrey Heinz. 2020. Computing and classifying reduplication with 2-way finite-state transducers. *Journal of Language Modelling*, 8(1):179–250.

J. A. Green. 1951. On the structure of semigroups. *Annals of Mathematics*, 54:163.

Jeffrey Heinz. 2010. String extension learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 897–906, Uppsala, Sweden.

Sharon Inkelas and Laura J Downing. 2015. What is reduplication? typology and analysis part 1/2: The typology of reduplication. *Language and linguistics compass*, 9(12):502–515.

Andreas Krebs, Kamal Lodaya, Paritosh K Pandya, and Howard Straubing. 2020. Two-variable logics with some betweenness relations: Expressiveness, satisfiability and membership. *Logical Methods in Computer Science*, 16.

Michal Kunc and Alexander Okhotin. 2011. Describing periodicity in two-way deterministic finite automata using transformation semigroups. In *International Conference on Developments in Language Theory*, pages 324–336. Springer.

Dakotah Lambert. 2022. *Unifying Classification Schemes for Languages and Processes with Attention to Locality and Relativizations Thereof*. Ph.D. thesis, State University of New York at Stony Brook.

Dakotah Lambert. 2026. Multitier phonotactics with logic and algebra. *Phonology*, 43.

Dakotah Lambert and Jeffrey Heinz. 2023. An algebraic characterization of total input strictly local functions. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2023*, pages 25–34, Amherst, MA.

Dakotah Lambert and Jeffrey Heinz. 2024. Algebraic reanalysis of phonological processes described as output-oriented. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2024*, pages 129–138, Irvine, CA.

John J. McCarthy and Alan Prince. 1995. Faithfulness and reduplicative identity. In Jill N. Beckman, Laura Walsh Dickey, and Suzanne Urbanczyk, editors, *Papers in Optimality Theory*. Graduate Linguistic Student Association, University of Massachusetts, Amherst, MA.

Robert McNaughton. 1974. Algebraic decision procedures for local testability. *Mathematical Systems Theory*, 8(1):60–76.

Edith A. Moravcsik. 1978. Reduplicative constructions. In Joseph H. Greenberg, Charles A. Ferguson, and Edith A. Moravcsik, editors, *Universals of Human Language. Volume 3: Word Structure*, pages 297–334. Stanford University Press, Stanford, CA.

Anil Nerode. 1958. Linear automaton transformations. *Proceedings of the American Mathematical Society*, 9(4):541–544.

Eric Raimy. 2000. *The phonology and morphology of reduplication*. Walter de Gruyter.

Brian Roark and Richard Sproat. 2007. *Computational approaches to morphology and syntax*. OUP Oxford.

James Rogers and Dakotah Lambert. 2019. Some classes of sets of structures definable without quantifiers. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 63–77, Toronto, Canada.

Carl Rubino. 2005. Reduplication: Form, function and distribution. *Studies on reduplication*, 28:11–29.

Carl Rubino. 2013. Reduplication. In Matthew S. Dryer and Martin Haspelmath, editors, *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Marcel Paul Schützenberger. 1965. On finite monoids having only trivial subgroups. *Information and Control.*, 8(2):190–194.

Thomas Schwentick, Denis Thérien, and Heribert Vollmer. 2001. Partially-ordered two-way automata: A new characterization of DA. In *International Conference on Developments in Language Theory*, pages 239–250. Springer.

Denis Thérien and Thomas Wilke. 1998. Over words, two variables are as powerful as one quantifier alternation. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, pages 234–240.

Yang Wang. 2021. Recognizing reduplicated forms: Finite-state buffered machines. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 177–187.

Appendix: Limitations

The chief formal limitation of this paper is that, to my knowledge, there is not yet any notion of a canonical form of a 2DFT, at least under the standard semantics. In the world of one-way machines, an automaton is called *minimal* if its states correspond exactly to the Nerode equivalence classes over its language (Nerode, 1958). This extends directly to transducers. Although Lambert (2022) takes a minimal 1DFT with the sink state removed (if necessary) to be canonical, there is no canonical two-way transducer. This means that any semigroup analysis of a two-way machine is only sufficient to give an upper bound on complexity—for instance, given that a particular two-way machine is in $\mathbb{D}\mathbf{A}$, we only know that the function computed is *at most* $\mathbb{D}\mathbf{A}$.

While Bojańczyk (2014) proves the existence of a canonical device for transductions with origin information, how to apply this result to the 2DFTs in RedTyp is not straightforward. Without a simple way to automatically minimize the machines in RedTyp, any improvements to the machines were made by hand. It is important to note that this can have an effect on their classification. For example, it was noted in Section 3.2 that the transducer for Initial-C reduplication in Agta given in Figure 2 was adapted slightly from RedTyp. Two changes were made: 1. In RedTyp, the direction of the transitions from q_1 to q_2 is mistakenly positive (only in the transducer code; it is correct in the diagram), and 2. There was a self-loop on q_2 with the label $(\Sigma, \lambda, +1)$ that was removed. While that loop does not change the function (because it is impossible to read any alphabet symbols in q_2), it changes the semigroup. Compare the semigroup structure given in 1 to the semigroup of the transducer with the superfluous loop, given in Table 2.

•	c	v	cv	vc
c	c	cv	cv	c
v	vc	v	v	vc
cv	c	cv	cv	c
vc	vc	v	v	vc

Table 2: Agta semigroup, with extra loop

Although the semigroup in Table 2 is in both $\mathbb{D}\mathbf{A}$ and $\mathbb{L}\mathbf{J}_1$, there is not much else to say about it. However, removing the loop on q_2 yields the semigroup in Table 1, which belongs to the more constrained classes $\mathbf{K}$ and $\mathbb{L}\mathbf{1}$. In addition to adjusting the transducer for Agta, I did the same for the seven other languages which have identical transducers, as well as for the transducer for Chinanteco final-C reduplication (as labeled in the appendix). These were done by visual inspection for extraneous loops, but because we lack canonical 2DFTs, future work may in principle find smaller machines with simpler semigroups for any of these machines.

A Appendix: Tables of Results

This appendix contains tables that summarize the results from the study of RedTyp. This appendix also includes the classes $\mathfrak{L}$-Trivial and $\mathfrak{R}$-Trivial, which are properly contained between $\mathbf{D}$ and $\mathbb{D}\mathbf{A}$ and between $\mathbf{K}$ and $\mathbb{D}\mathbf{A}$, respectively. These are classes related to the $\mathfrak{L}$ and $\mathfrak{R}$ relations defined originally by Green (1951). Interested readers are referred to (Lambert, 2022; §4). Table 3 summarizes those machines that do not have semigroups in $\mathbb{L}\mathbf{1}$. Table 4 contains only those transducers that were edited to remove extraneous loops and lower their complexity. Table 5 contains those processes that were in $\mathbb{D}\mathbf{A}$, $\mathbb{L}\mathbf{J}_1$, $\mathbb{L}\mathbf{1}$, but no smaller classes. Lastly, Table 6 contains all processes that are trivial ($\mathbf{1}$), and thereby belong to all classes discussed in this paper.

#	Language	Morpheme ID	$\mathbb{D}$**A**	$\mathbb{L}\mathbf{J}_1$	$\mathbb{L}$**1**	$\mathfrak{L}$-Triv
1	Armenian	arm-ECHO-Echo	✓	✓		✓
2	English	eng-derisive-Echo	✓	✓		✓
3	Turkish	tur-ECHO-Echo	✓	✓		✓
4	Tuvan	Tuvan-ECHO-Echo	✓	✓		✓
7	Kannada	kan-ECHO-Echo	✓	✓		✓
8	Kolami	Kolami-ECHO-Echo	✓	✓		✓
9	Tamil	tam-ECHO-Echo	✓	✓		✓
58	Bikol	bik-IMPF-InitialCV	✓	✓		
59	Ewe	ewe-CONT/NMNZ?-InitialCV	✓	✓		
60	Tagalog	tgl-unclear-InitialCV	✓	✓		
61	Tohono O'odham	TohonoOodham-PL-InitialCV	✓	✓		
68	Oykangand	Oykangand-unclear-InitialCVC	✓	✓		
89	Mangarayi	Mangarayi-PL/INT-InitialVC	✓	✓		
91	Chamorro	cha-CONT-Stressed	✓	✓		
94	Samoan	smo-PL-PenultimateCV	✓	✓		
74	Ilocano	ilo-PL/PROG-InitialCVC	✓			
98	Malay	may-unclear-Total	✓			

Table 3: Processes that are not $\mathbb{L}$**1**

#	Language	Morpheme ID	$\mathbb{D}$**A**	$\mathbb{L}\mathbf{J}_1$	$\mathbb{L}$**1**	$\mathfrak{L}$-Triv	**D**	$\mathfrak{R}$-Triv	**K**
10	Chinanteco	Chinanteco-PLPoss-FinalC	✓	✓	✓	✓	✓		
35	Agta	Agta-DIM-InitialC	✓	✓	✓			✓	✓
36	Bikol	bkl-polysemous-Initial C	✓	✓	✓			✓	✓
37	Marshallese	mah-ADJZ-InitialC	✓	✓	✓			✓	✓
38	Nupe	Nupe-unclear-Initial C	✓	✓	✓			✓	✓
39	Pacoh	Pacoh-denominal verbalization-Initial C	✓	✓	✓			✓	✓
40	Shilh	Shilh-PROG-InitialC	✓	✓	✓			✓	✓
41	Yakan	Yakan-repeated/distributed-Initial C	✓	✓	✓			✓	✓
42	Yoruba	yor-polysemous-Initial C	✓	✓	✓			✓	✓

Table 4: Processes that had extraneous loops removed

#	Language	Morpheme ID	#	Language	Morpheme ID
77	Agta	Agta-PL-InitialCVC	24	Mokilese	Mokilese-NMNZ-FinalCVC
43	Akan	aka-unclear-InitialCV	67	Mokilese	Mokilese-unclear-InitialCVC
20	Ao Naga	AoNaga-DISTR-FinalCVC	75	Mokilese	Mokilese-PROG-InitialCVC
13	Arrente	Arrente-INSTR-FinalCV	81	Ngiyambaa	Ngiyambaa-unclear-InitialCVCV
44	Attic Greek	grc-PERF-InitialCV	47	Nitinaht	Nitinaht-resemble-InitialCV
62	Aztek	Aztek-xbyx-InitialCV	23	Pacoh	Pacoh-DISTR-FinalCVC
69	Aztek	Aztek-PROG-InitialCVC	64	Papago	Papago-PL-InitialCV
84	Balangao	Balangao-unclear-InitialCVCV	26	Paumari	Paumari-unclear-FinalCVCV
56	Bikol	bkl-unclear3-InitialCV	55	Pima	Pima-PL-Initial CV
11	Chamorro	cha-INT-FinalCV	30	Quileute	Quileute-FREQ/DISTR-FirstCafterFirstV
93	Chuckchee/Koryak	Chuckchee/Koryak-ABS-NonlocalInitialCVC	63	Quileute	Quileute-PL-InitialCV
66	Chumash	Chumash-unclear-InitialCVC	48	Rotuman	Rotuman-CONT,FREQ,HAB?-InitialCV
92	Creek	mus-PL-NonlocalInitialCV	72	Samala/Chumash	Samala/Chumash-REP-InitialCVC
18	Dakota	dak-VRBZ-FinalCVC	33	Semai	Semai-unclearFirstLastC
78	Dyirbal	Dyirbal-moreX-InitialCVCV	27	Siriono	Siriono-CONT-FinalCVCV
85	Dyirbal	Dyirbal-unclear-InitialCVCV	73	Somali	som-PL-InitialCVC
79	Fox	Fox-CONT?-InitialCVCV	76	Southern Paiute	SouthernPaiute-unclear-InitialCVC
14	Hausa	hau-PL-FinalCV	49	Sundanese	sun-CONT-InitialCV
45	Hausa	hau-INT/PLUR-InitialCV	50	Sundanese	sun-pretend-InitialCV
15	Hiaki/Yaqui	Hiaki-HAB-InitialCV	51	Sundanese	sun-transitivize-InitialCV
16	Hopi	Hopi-unclear-FinalCV	25	Tagalog	tgl-DIM-FinalCVCV
70	Ilocano	ilo-IMPF-InitialCVC	52	Tagalog	tgl-NMNZ-InitialCV
19	Kaingang	Kaingang-PL-FinalCVC	53	Tagalog	tgl-unclear2-InitialCV
82	Kinande	Kinande-unclear-InitialCVCV	57	Tagalog	tgl-unclear3-InitialCV
17	Kwaza	Kwaza-PROG?-FinalCV	65	Tagalog	tgl-IMPF-InitialCV
31	Levantine Arabic	ara-INT-FirstCafterSecondC	86	Tagalog	tgl-unclear-InitialCVCV
80	Limos Kalinga	LimosKalinga-ITER	34	Temiar	Temiar-ACT/CONT-FirstLastC
46	Lushootseed	Lushootseed-DIM-InitialCV	28	Thao	Thao-INT-FinalCVCV
71	Lushootseed	Lushootseed-DISTR-InitialCVC	54	Tsimshian	tsi-unclear-InitialCV
88	Lushootseed	Lushootseed-OutofControlEtc-InitialVC	29	Tzeltal	Tzeltal-unclear-FinalVC
83	Makassarese	Makassarese-unclear-InitialCVCV	87	Warlpiri	Warlpiri-unclear-InitialCVCV
12	Manam	Manam-polysemous-FinalFoot	90	Washo	was-PL-InitialVCV
21	Marshallese	mah-VBZ-FinalCVC	32	Zuni	zun-REP-FirstCafterSecondC
22	Mokilese	Mokilese-VRBZ-FinalCVC			

Table 5: Processes that are $\mathbb{D}\mathbf{A}$, $\mathbb{L}\mathbf{J}_1$, and $\mathbb{L}\mathbf{1}$

#	Language	Morpheme ID	#	Language	Morpheme ID
95	Amele	Amele-ITER-Total	112	Pacoh	Pacoh-RECP-Total
96	Boumaa Fijian	BoumaaFijian-INT-Total	131	Shipibo	Shipibo-CONT-Triplication
102	Dholuo	Dholuo-mitigation-Total	113	Somali	som-toXAtAllSides-Total
103	Dyirbal	Dyirbal-PL-Total	114	Sundanese	sun-INT-Total
104	Dyirbal	Dyirbal-RECP-Total	115	Sundanese	sun-notEvenX-Total
129	Emai	Emai-ADVZ-Triplication	116	Sundanese	sun-RECP-Total
105	Ewe	ewe-XbyX-Total	117	Swahili	swa-DIM-Total
5	Hindi	hin-ECHO-Echo	119	Tagalog	tgl-everyX-Total
106	Hungarian	hun-occassionally-Total	118	Tausug	Tausug-unclear-Total
99	Indonesian	ind-INT/REP-Total	132	Telugu	tel-emphasis?-Triplication
100	Indonesian	ind-RECP-Total	120	Thai	tha-DIM-Total
107	Indonesian	ind-PL-Total	121	Thai	tha-INT-Total
108	Kanuri	Kanuri-glossonym-Total	122	Twi	twi-DenomADJZ-Total
6	Kashmiri	kas-ECHO-Echo	133	twi	twi-ADVZ-Triplication
109	Malay	may-similar-Total	123	Tzeltal	tzeltal-CONT-Total
97	Mandarin	chi-INT?-Total	124	Tzeltal	Tzeltal-DISTR-Total
101	Mandarin	chi-every-Total	125	Tzeltal	Tzeltal-veryMuch-Total
110	Mokilese	Mokilese-PROG-Total	126	Yami	Yami-RECP-Total
130	Mokilese	Mokilese-CONT-Triplication	127	Yessan-Mayo	YessanMayo-DISTR-Total
111	Nez Perce	NezPerce-DIM-Total	128	Yoruba	yor-PL-Total

Table 6: Processes that are trivial ($\mathbf{1}$)

Learning reduplicative templates as hidden structures: the case of reduplication-phonology interactions

Yang Wang
University of Utah
yangx.wang@utah.edu

1 Defining the problem

Models of morphophonological learning have focused primarily on concatenative processes, leaving the challenges of non-concatenative morphology largely unaddressed. Reduplication, the systematic copying operation (e.g., Ilokano pluralization [kal-kaldíŋ] 'goats'), is particularly revealing because successful learning requires the joint inference of prosodic templates that govern copying, underlying representations (URs) of stems and other affixes, as well as the phonological grammar.

Consider the schematic language in Table (1). The challenge of moving from the surface forms in Table (1a) to the analysis in Table (1b) lies in the mutual dependence between the grammar and the hidden structures. Solely based on the CAT paradigm, the learner cannot determine whether the reduplicative morpheme is a heavy syllable or a prosodic word, nor whether the stem UR is /bet/ with voicing assimilation or /bed/ with final devoicing.

Meanings	*Forms*
CAT_{STEM}	bet
CAT-PL	bed-a
CAT-DIM	bed-bet
DOG_{STEM}	panat
DOG-PL	panat-a
DOG-DIM	panat-nat

(a) Surface forms as input.

Morphemes	*Hidden units*
CAT_{STEM}	/bed/
DOG_{STEM}	/panat/
PL	/-a/
DIM	$\text{RED} = \sigma_{\mu\mu}$
Phonology: d → t / _#	

(b) The analysis.

Table 1: A schematic language with word-final devoicing and suffixing heavy-syllable reduplication.

The DOG paradigm provides the crucial disambiguating evidence. If the reduplicative template is the whole word, the predicted DOG-DIM form would be *[panat-panat], while a heavy-syllable template yields [panat-nat], which is what surfaces. The choice between /bed/ and /bet/ for CAT works analogously: positing /panat/ for DOG predicts [panat-a] for DOG-PL, which is observed; positing intervocalic voicing instead would wrongly predict *[panad-a]. Hence the CAT alternation must reflect devoicing of an underlying /bed/.

Within the typology of reduplication, Table (1) illustrates only one attested pattern of interaction between reduplication and segmental alternations, namely *normal application*, in which the process applies in all and only the expected environments. Other attested interactions include *overapplication* and *underapplication* (Wilbur, 1973); see McCarthy and Prince (1995) and Wang (2024) for plausible real language examples. Table 2 contrasts the three patterns: in *over*application (2a), devoicing applies to the first copy despite the absence of the conditioning environment. In *under*application (2b), devoicing fails to apply despite the presence of a word-final conditioning context.

Meanings	*Forms*
CAT_{STEM}	bet
CAT-PL	bed-a
CAT-DIM	bet-bet

(a) Lg2. *Overapplication*.

Meanings	*Forms*
CAT_{STEM}	bet
CAT-PL	bed-a
CAT-DIM	bed-bed

(b) Lg3. *Underapplication*.

Table 2: Typology of reduplication–phonology interactions. Only the CAT paradigm is shown; the DOG paradigm is identical to Table 1 in all three languages.

Prior computational work on reduplication has treated it as a morphology problem while setting aside its interactions with the rest of the phonological grammar (Vania and Lopez, 2017; Wilson, 2018; Dolatian and Heinz, 2018; Xu et al., 2020; Nelson et al., 2020; Beguš, 2021; Todd et al., 2022; Ellis et al., 2022). But the three attested patterns of reduplication–phonology interactions are precisely where copying and segmental phonology cannot be learned in isolation: each requires the learner to resolve hidden-structure ambiguities about templates

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 460–463
July 3-4, 2026 ©2026 Association for Computational Linguistics

and URs jointly with the grammar that relates them. We present a learner that performs this joint inference, a combination that, to our knowledge, has not been directly modeled in prior work. We show that it captures the attested typology.

Our goal is to model phonological acquisition: given a small paradigmatic input of the kind a child is likely to encounter, what hypotheses about underlying representations, templates, and grammar must the learner entertain in order to converge on the attested patterns? The EM-MaxEnt framework (O'Hara, 2017; Wang and Hayes, to appear) is well-suited to this question because (i) the constraints encode substantive linguistic hypotheses, namely, Base-Reduplicant Correspondence Theory (BRCT) (McCarthy and Prince, 1995) and Prosodic Morphology (McCarthy and Prince, 1986) whose learnability is itself the object of study; and (ii) the learned weights and posterior distributions over hidden structures are directly interpretable as theoretical claims, supporting analysis of what the learner does and why.

2 The model

2.1 The EM-MaxEnt framework

Wang and Hayes (to appear) propose a general framework for jointly learning URs and segmental phonology, in which URs are represented as latent categorical variables with multinomial distributions over candidates drawn from observed allomorph sets. This model, however, is limited to concatenative processes. To extend it to reduplication, we treat the reduplicative morpheme's prosodic template as a latent representation of the same formal type as a segmental UR: for each morpheme m the learner maintains a categorical prior θ_m over a finite set $\mathrm{UR}(m)$ of candidate representations. For a segmental morpheme, $\mathrm{UR}(m)$ is an allomorph set; for the reduplicative morpheme DIM, $\mathrm{UR}(\textsc{dim}) = \{\sigma_\mu, \sigma_{\mu\mu}, \mathrm{Ft}, \mathrm{PrWd}\}$. The same EM machinery learns both.

The probability of a surface form s given a meaning ω is the mixture

$$P(s \mid \omega; W, \theta) = \sum_{u \in \mathrm{UR}(\omega)} P(s \mid u; W) \prod_{m \in \omega} \theta_m(u_m), \tag{1}$$

where $u = (u_m)_{m \in \omega}$ ranges over joint UR assignments. The MaxEnt formula for each (UR, SR) pair is as in (2).

$$P(s \mid u; W) = \frac{exp\big(-\sum_k w_k\, C_k(u, s)\big)}{\sum_{s'} exp\big(-\sum_k w_k\, C_k(u, s')\big)} \tag{2}$$

Learning maximizes the regularized log conditional likelihood $\mathcal{L}(W, \theta) = \ln P(D \mid W, \theta) - \sum_i \frac{(w_i - \mu_i)^2}{2\sigma_i^2}$ via Expectation-Maximization (Dempster et al., 1977): the E-step computes $E(u, s, \omega) \propto P(s \mid u; W) \prod_m \theta_m(u_m)$; based on the computed expected counts, the M-step re-estimates W via L-BFGS-B and each θ_m by normalized expected counts. We use $2\sigma_i^2 = 10^5$ throughout, with non-negative weight constraints, and terminate when $\Delta\mathcal{L} < 10^{-3}$.

2.2 The reduplication learning component.

We augment the framework with three major components, the first of which is the *template space* as above.

Constraint set We use 12 standard constraints from BRCT (McCarthy and Prince, 1995) and related OT literature, supported by empirical evidence (Siah et al., under review); these include input-base identity, identity between copies, anchoring, and contiguity constraints. We additionally propose RED=X, a single constraint in which X is bound at evaluation time by the template UR under consideration in the current (UR, SR) pair. It assigns a violation when the reduplicant in the surface form does not have the prosodic shape X specified by the input pair. For example, paired with the template UR $\sigma_{\mu\mu}$, a candidate reduplicant [be] violates RED=X, while [bet] does not. This diverges from standard BRCT, where each prosodic template corresponds to a separate constraint; the single weighted RED=X acts as a gatekeeper controlling the extent to which the posited template influences the reduplicant's shape.

GEN Candidate surface forms are constructed in three stages: ***reduplicant generation*** (each base segment may or may not surface, yielding 2^n candidates; e.g., for /bed/: {bed, be, ed, bd, b, e, d, ∅}), ***reduplicant placement*** (prefixing, suffixing, infixing, back-copying), and ***alternation substitution*** (each candidate is expanded with segmental alternants from the regular phonology; e.g. if [bed-bed] is in the candidate set, a [t]∼[d] alternation would also produce [bed-bet], [bet-bed] and [bet-bet]). Although each toy language contains only six

observed surface forms, the learner evaluates 1,992 hidden-structure/surface-form candidate pairs.

3 Evaluation and results

We evaluate the learner on the three schematic languages in Tables 1 and 2. The hypothesis space is identical across runs, same four candidate templates, same UR candidates from observed allomorphs, and the same 13 constraints. Only the CAT-DIM form differs, and this difference drives the learner to different grammars.

At initialization, all θ_m are uniform and constraint weights are zero. Across all three languages, the learner recovers the alternating-stem UR /bed/ and the template $\sigma_{\mu\mu}$ with posterior probability $> .999$. For Lg1 and Lg2, the learned grammar assigns probability $> .999$ to the observed outputs. For Lg3, the data are consistent only with a variable grammar, and the learner converges on a 0.50/0.50 distribution that frequency-matches the input. Table 3 presents the learned weights of key constraints across the three languages.

Constraint	Lg1 *normal*	Lg2 *over*	Lg3 *under*
*FINALVOICEDOBS	17.32	22.19	**8.56**
IDENT-IO[VOICE]	9.35	13.38	**8.56**
*VTV	1.34	6.99	1.16
*MARGINCLUSTER	6.28	3.53	6.22
MAX-IO	11.21	7.7	11.20
IDENT-BR[VOICE]	0.05	21.8	15.92
MAX-BR	6.66	0	6.70
CONTIG-BR	10.45	9.32	10.16
L-ANCHOR-BR	0.39	0.00	0.00
R-ANCHOR-BR	9.02	21.64	8.62
ALIGNREDL	0.00	0.00	0.00
ALIGNREDR	5.10	7.69	5.16
RED=X	21.44	7.34	21.18

Table 3: Learned constraint weights for the three languages. In Lg3, *FINALVOICEDOBS and IDENT-IO[VOICE] converge to equal weights, yielding free variation.

In Lg1 (*normal* application), the training datum [bed-bet] already exhibits different voicing across copies, so the learner has no pressure to push IDENT-BR[VOICE] up; it stays near zero ($w = 0.05$). Devoicing applies independently to each copy via *FINALVOICEDOBS outweighing IDENT-IO.

In Lg2 (*over*application), [bet-bet] forces the learner to explain why /bed/ surfaces as [bet] in the left copy. Devoicing in the left copy has no markedness motivation; the only available explanation in the hypothesis space is that BR identity pulls the base toward the markedness-driven [t] of the reduplicant. EM accordingly raises IDENT-BR to $w = 21.8$, with *FINALVOICEDOBS at $w = 22.19$ providing markedness pressure on the second copy.

In Lg3 (*under*application), the training data are mutually incompatible under any categorical grammar in the hypothesis space. The bare stem [bet] requires *FINALVOICEDOBS to outweigh IDENT-IO; the reduplicated form [bed-bed] requires the opposite. The learner settles to *frequency match*: *FINALVOICEDOBS and IDENT-IO converge to equal weights ($w = 8.56$ each), with IDENT-BR high ($w = 15.92$) to enforce identity between copies. The resulting grammar predicts 0.50/0.50 free variation between [bed]∼[bet] for CAT_{STEM} and [bed-bed]∼[bet-bet] for CAT-DIM, as in Table 4.

	*FINALVOICEDOBS 8.56	ID-IO 8.56	ID-BR 15.92	*H*	*P*
Input: /bed/					
[bed]	1			8.56	**.500**
[bet]		1		8.56	**.500**
Input: /bed-RED/					
[bed-bet]			1	15.92	.000
[bed-bed]	1			8.56	**.500**
[bet-bet]		1		8.56	**.500**
[bet-bed]	1	1	1	33.04	.000

Table 4: Underapplication grammar: equal weights on *FINALVOICEDOBS and ID-IO yield free variation. Candidates with BR mismatches lose categorically.

This recapitulates the classical asymmetry of McCarthy and Prince (1995, p5) as an emergent property of the hypothesis space. Under BRCT, categorical underapplication requires an independently motivated constraint to break the tie. Without such a constraint, no grammar in the space prefers [bed-bed] categorically over [bet-bet], and the learner converges on the free-variation grammar. Although it remains unclear how humans acquire underapplication, some documented underapplication processes do seem to involve free variation (see Tagalog in Zuraw, 2002).

4 Discussion

As a first step, the model demonstrates that EM-MaxEnt inference over multiple interacting hidden structures, both prosodic templates and segmental URs, is feasible and effective for learning

reduplication-phonology interactions. While the underlying optimizer is the EM-MaxEnt framework of Wang and Hayes (to appear), the contribution here is in the *hypothesis space*: representing prosodic templates as URs of the reduplicative morpheme, introducing RED=X as the mechanism by which templates exert grammatical influence, and constructing GEN to generate reduplication candidates parametrically.

The asymmetry between overapplication and underapplication arises from this hypothesis-space structure. Overapplication is acquired categorically when the data leave only BR-identity as an available explanation; underapplication, lacking an independently motivated tie-breaker, drives the learner to free variation.

The 1,992 candidate pairs here arise from the combinatorial structure of GEN. This grows with morpheme length and the alternation set rather than with lexicon size; adding more stems sharing the same affix without introducing new alternations gives the learner stronger evidence about that affix's UR rather than enlarging the search. Future work will evaluate the model on natural language reduplication systems and compare it to alternative frameworks for non-concatenative morphology such as Generalized Non-linear Affixation (Bermúdez-Otero, 2012). More broadly, the same EM-MaxEnt machinery extends from segmental URs to prosodic templates, suggesting a unified route to learning phonological hidden structures.

References

Gašper Beguš. 2021. Identity-based patterns in deep convolutional networks: Generative adversarial phonology and reduplication. *TACL*, 9:1180–1196.

Ricardo Bermúdez-Otero. 2012. The architecture of grammar and the division of labour in exponence. In Jochen Trommer, editor, *The Morphology and Phonology of Exponence*, pages 8–83. Oxford University Press, Oxford.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22.

Hossep Dolatian and Jeffrey Heinz. 2018. Learning reduplication with 2-way finite-state transducers. In *Proceedings of the 14th International Conference on Grammatical Inference*, volume 93 of *Proceedings of Machine Learning Research*, pages 67–80. PMLR.

Kevin Ellis, Adam Albright, Armando Solar-Lezama, Joshua B Tenenbaum, and Timothy J O'Donnell. 2022. Synthesizing theories of human language with bayesian program induction. *Nature Communications*, 13(1):5024.

John J. McCarthy and Alan S. Prince. 1986. Prosodic morphology. Ms., University of Massachusetts, Amherst, and Brandeis University, Waltham, Mass.

John J. McCarthy and Alan S. Prince. 1995. Faithfulness and reduplicative identity. In *Papers in Optimality Theory*, Amherst, MA. Graduate Linguistic Student Association, Dept. of Linguistics, University of Massachusetts.

Max Nelson, Hossep Dolatian, Jonathan Rawski, and Brandon Prickett. 2020. Probing RNN encoder-decoder generalization of subregular functions using reduplication. *Society for Computation in Linguistics*, 3(1).

Charlie O'Hara. 2017. How abstract is more abstract? Learning abstract underlying representations. *Phonology*, 34(2):325–345.

Jian Leat Siah, Sam Zukoff, and F. F. Hsieh. under review. Resolving reduplicative opacity in Malay nasal spreading: Argument for Base-Reduplicant correspondence theory. *Language*.

Simon Todd, Annie Huang, Jeremy Needle, Jennifer Hay, and Jeanette King. 2022. Unsupervised morphological segmentation in a language with reduplication. In *Proceedings of the 19th SIGMORPHON Workshop*, pages 12–22, Seattle, Washington. Association for Computational Linguistics.

Clara Vania and Adam Lopez. 2017. From characters to words to in between: Do we capture morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2016–2027, Vancouver, Canada. Association for Computational Linguistics.

Yang Wang. 2024. *Studies in morphophonological copying: Analysis, experimentation and modeling*. Ph.D. thesis, University of California, Los Angeles.

Yang Wang and Bruce Hayes. to appear. Learning phonological underlying representations: the role of abstractness. *Linguistic Inquiry*.

Ronnie Bring Wilbur. 1973. *The phonology of reduplication*. University of Illinois at Urbana-Champaign.

Colin Wilson. 2018. Modeling morphological affixation with interpretable recurrent networks: sequential rebinding controlled by hierarchical attention. In *CogSci*.

Hongzhi Xu, Jordan Kodner, Mitch Marcus, and Charles Yang. 2020. Modeling morphological typology for unsupervised learning of language morphology. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6672–6681.

Kie Zuraw. 2002. Aggressive reduplication. *Phonology*, 19(3):395–439.

Do Large Language Models Acquire Phrase-Based Processing? Evidence from Eye Movements and Model–Brain Alignment After Fine-Tuning

Xufeng Duan[1] **Zhengwu Ma**[2] **Zhaoqian Yao**[1] **Jixing Li**[2] **Zhenguang Cai**[1]
[1]The Chinese University of Hong Kong
[2]City University of Hong Kong
jixingli@cityu.edu.hk zhenguangcai@cuhk.edu.hk

Abstract

Autoregressive large language models (LLMs) process text token-by-token, yet the human language system operates over multi-word units. We ask whether aggregating LLM representations at the *phrase* level yields a closer correspondence to human reading behavior and language cortex than the default word-level representations, and whether phrase-segmentation fine-tuning amplifies this correspondence. Using Meta-Llama-3.1-8B (base and fine-tuned), we provide three converging lines of evidence. First, phrase-level attention features predict regressive eye-saccade patterns more closely than word-level features; a partial correlation analysis with a shuffled-boundary control indicates that this is not solely an aggregation artifact and that linguistic chunk boundaries explain unique variance beyond word-level attention. Second, fMRI encoding analyses show that fine-tuning selectively improves phrase encoding in left superior temporal gyrus and inferior frontal gyrus, with no improvement for word representations. Third, representational similarity analysis confirms a phrase-specific gain in model–brain geometric alignment. These results identify phrase-level representation as a critical granularity for LLM–human correspondence and suggest that targeted training can model human-like compositional processing, linking computational representations to hierarchical theories of language.

1 Introduction

Autoregressive large language models (LLMs) achieve strong performance by predicting the next token conditioned on preceding context (Radford et al., 2019). Because these models learn statistical regularities from text alone, they offer a powerful test bed for understanding what aspects of human language processing can emerge from distributional learning—and where the two systems diverge (Cai et al., 2024; Duan et al., 2024). However, a principled model–brain comparison requires choosing a representational granularity at which to extract and evaluate model features. Most prior work on model–brain alignment operates at the token or word level (Schrimpf et al., 2021; Goldstein et al., 2022; Caucheteux and King, 2022; Wu et al., 2025). Yet the human language system is organized around multi-word constituents: phrases are the primary units of planning, production, and comprehension (Garrett, 1975; Fedorenko et al., 2016; Hu et al., 2023; Gibson, 1998). This mismatch raises a fundamental question: at what representational scale do LLM internal states best correspond to human cognitive and neural processing?

The Granularity Problem Token-level representations are the default unit of analysis for transformer models, but they may not provide the most informative window into how models encode language-relevant structure. Effective next-token prediction often requires integrating information across multi-word spans (Manning et al., 2020). For instance, determiner selection ("a" vs. "an") depends on upcoming phonological context. More strikingly, subject–verb agreement can survive substantial intervening material: speakers and comprehenders routinely maintain agreement across prepositional phrases, relative clauses, and other syntactic interpolations (e.g., "The key to the cabinets *is*..."), and errors in such configurations, so-called broken agreement, reveal that the language system tracks hierarchical phrase structure rather than simple linear adjacency (Bock and Miller, 1991). These observations suggest that multi-word phrases, not individual tokens, are the natural grain at which structural information is maintained. The question is therefore not whether models *explicitly compute* phrases, but whether *phrase-level aggregation of model representations* provides a better window into the structure that the human language system tracks.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 464–476
July 3-4, 2026 ©2026 Association for Computational Linguistics

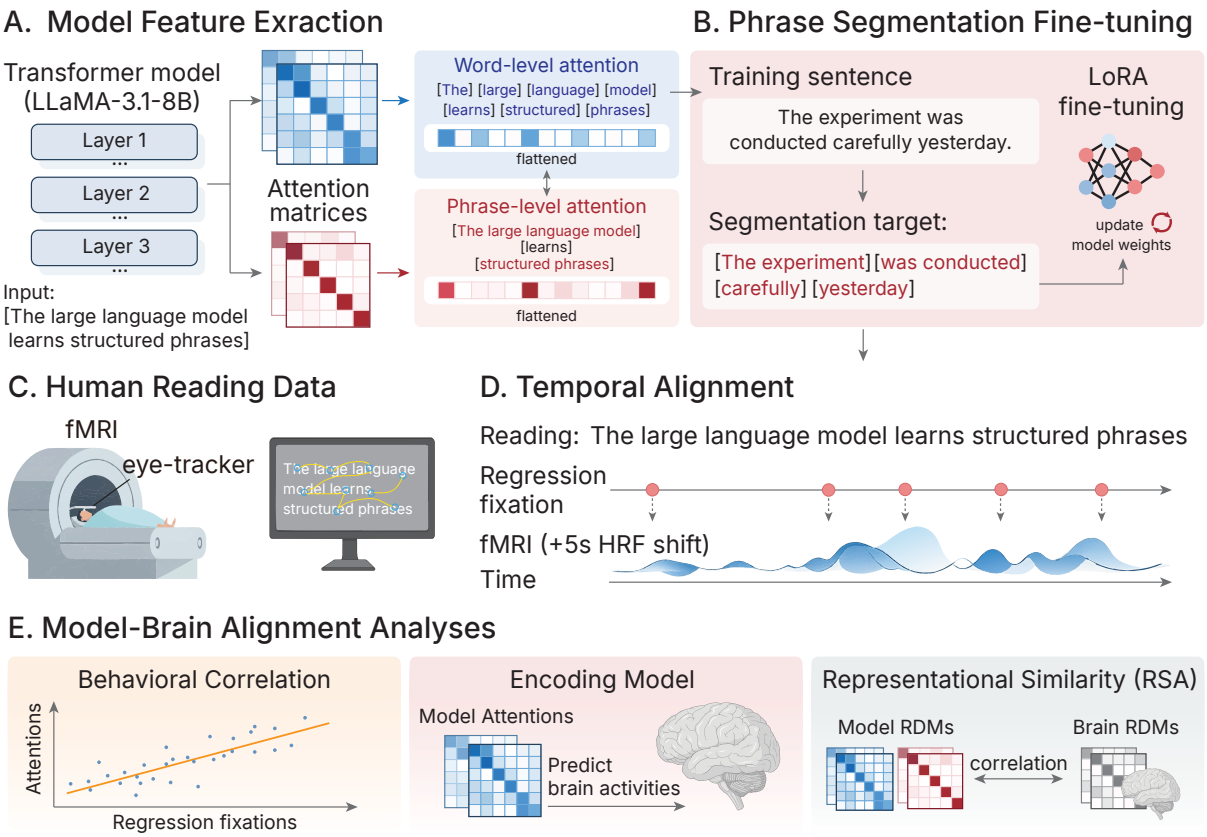


Figure 1: Methodological framework. **(A–B)** Attention matrices from Llama-3.1-8B are extracted at word-level and phrase-level granularities. **(C)** Eye-tracking fixation onsets are mapped to fMRI volumes (+5 s hemodynamic lag). **(D–E)** Encoding models and LOSO cross-validated cluster permutation tests identify brain regions with significant model alignment.

Phrase-Level Structure in Human Language Processing Psycholinguistic and neuroscientific evidence indicates that humans process language using structured, multi-word units (Fedorenko et al., 2016; Hu et al., 2023). Eye movements during reading reflect incremental difficulty and predictability effects that operate over spans larger than individual words: fixation durations and saccade patterns are modulated by phrase-level predictability and constituent boundaries, not only by properties of the currently fixated word (Rayner and Duffy, 1986; Smith and Levy, 2013). Neuroimaging studies consistently implicate left temporal and inferior frontal regions in combinatorial and sentence-level computations, with activity in these regions scaling with the size and complexity of phrasal constituents (Pallier et al., 2011; Fedorenko et al., 2011; Hu et al., 2023). If phrase-level aggregation captures information that is more closely aligned with the granularity at which the brain organizes language processing, it should yield better correspondence between model features and both behavioral and neural measures.

The Interface Hypothesis We propose that phrase-level aggregation of LLM attention provides a better correspondence to human cognition than word-level attention, regardless of whether the model explicitly computes phrases. On this view, the phrase is the representational scale at which model internal states most closely correspond to human reading dynamics and neural language processing. The claim is not that models "know about" phrases in a symbolic sense, but rather that extracting model features at the phrase grain reveals structure that is otherwise obscured by token-level analysis. We use *phrase* in an operational, surface-level sense: our segmentation corresponds to what NLP calls *chunking* (Abney, 1991), non-overlapping multi-word groupings such as base noun phrases, verb groups, and prepositional/adverbial spans, rather than full hierarchical constituents of generative syntax. This usage is close in spirit to "construction"-level units in construction grammar (Goldberg, 2003), is consistent with psycholinguistic descriptions of chunks as units of incremental processing, and is robust

across syntactic frameworks. We return to this choice in the Discussion and Limitations.

The Fine-Tuning Leverage If the phrase-level interface can be strengthened, then explicitly training a model to recognize phrase boundaries should *amplify* the alignment between model representations and human neural activity. Critically, we frame this as amplification of an existing correspondence rather than creation of a new computational mechanism: the base model may already encode phrase-relevant structure, and fine-tuning may sharpen this structure toward the granularity at which the brain operates (Manning et al., 2020; Tenney et al., 2019).

Present Study We test the interface hypothesis and the fine-tuning leverage using Meta-Llama-3.1-8B as a base model and a parameter-efficient fine-tuned variant trained to produce explicit phrase segmentations. We evaluate both models on the Reading Brain Project naturalistic reading dataset, which provides simultaneous eye tracking and fMRI during text comprehension. Our central comparison contrasts features derived from attention at multiple granularities: a *word* condition based on token/word-level attention, a *phrase* condition obtained by aggregating attention within and between syntactically defined phrase spans, a *shuffled-phrase* control that preserves multi-word aggregation but disrupts linguistic boundaries, and a *phrase* − *word* residual that isolates compositional structure beyond lexical processing. We ask: (1) does phrase-level aggregation better predict human reading behavior, and does this advantage survive after controlling for multi-word aggregation? (2) does phrase-segmentation fine-tuning selectively enhance neural encoding of phrase representations? and (3) does fine-tuning improve the representational geometry alignment between model attention and human neural responses in language-related region? Figure 1 provides an overview of our experimental framework.

Our results reveal a robust phrase superiority effect in behavioral alignment, selective amplification of phrase-level neural encoding after fine-tuning, and convergent geometric evidence, while neither model encodes phrase-minus-word residual structure, constraining phrases as holistic representational units.

2 Method

2.1 Base Model

We use Meta-Llama-3.1-8B as our base model (Grattafiori et al., 2024), analyzing internal attention representations across all 32 transformer layers. We focus on attention weights rather than hidden states because our central comparison is with word-pair regressive-saccade matrices, which have the same pairwise structure as attention matrices and were the basis of prior model–brain alignment work on this dataset (Gao et al., 2025).

2.2 Phrase Segmentation Task

To test whether explicitly teaching chunk boundaries affects cognitive alignment, we fine-tuned the model on a supervised phrase-segmentation task.

Training data and format. We constructed a dataset of 200,000 sentences from BLiMP (Warstadt et al., 2019) (∼60,000) and SNLI (Bowman et al., 2015) (∼140,000), filtered for well-formedness and length (10–80 tokens). Each sentence was annotated by a deterministic rule-based pipeline applied to spaCy dependency parses (*en_core_web_trf*); the rules group tokens into base noun phrases, verb groups, and prepositional/adverbial spans (full rule set in Appendix A). The model takes a sentence and outputs the segmented version with chunk boundaries marked by pipes (e.g., "The quick brown fox | jumps | over the lazy dog | ."). We adopt a shallow chunking scheme for two reasons: (i) it provides robust, framework-neutral groupings; (ii) it mirrors the "chunking" level used in psycholinguistic studies of incremental processing.

2.3 Fine-Tuning Implementation

We fine-tuned the model using LoRA ($r = 64$, $\alpha = 128$) applied to all attention projection matrices (Hu et al., 2022), trained for 2 epochs with the TRL SFTTrainer (full hyperparameters in Appendix A). The fine-tuned model achieves 94.2% exact-string match against this rule-based gold on a held-out test set of 140 testing sentences from Reading Brain Project datasets, vs. 25.7% for the base model.

2.4 Dataset: Reading Brain Project Corpus

We evaluated model–human alignment using the Reading Brain Project corpus (Li and Clariana, 2019; Follmer et al., 2018; Hsu et al., 2019): simultaneous eye-tracking and fMRI from 50 participants reading naturalistic STEM articles (∼29.6

sentences/article, ∼10.3 words/sentence) in the scanner at TR = 0.4 s. Analyses focused on a left-hemisphere language mask (see Neural Encoding below).

2.5 Attention Extraction Conditions

For each sentence, layer j, and head k, we extracted the lower-triangular part (excluding the diagonal) of the $n_{\text{word}} \times n_{\text{word}}$ attention matrix—the right-to-left attention weights, consistent with the model's unidirectional structure.

Word condition. Word-level attention as above; subword-to-word alignment summed over "to" subwords and averaged over "from" subwords (Manning et al., 2020). For eye-movement analyses, we max-pooled across heads at each layer before extracting the lower triangle.

Phrase condition. Tokens belonging to the same chunk were grouped using the same rule-based segmentation, and attention and saccade matrices were pooled to chunk boundaries with identical aggregation; eye-movement correlations were computed in chunk-pair space.

Shuffled-phrase control. Phrase boundary positions were randomly permuted within each sentence, preserving the number and size of phrases; the same pooling was applied. This matches the aggregation granularity but disrupts linguistic groupings.

Phrase − Word condition. Residual obtained by subtracting word-level from phrase-level attention values (neural encoding and RSA only).

2.6 Eye Movements Alignment

We follow the analysis pipeline of Gao et al. (2025): both unidirectional LLM attention and regressive (right-to-left) eye saccades occupy lower-triangular pair space, so the two can be compared on a sentence-by-sentence basis.

Regression target. For each sentence and each participant, we built a matrix $E^{(s)} \in \mathbb{R}^{n^{(s)}_{\text{word}} \times n^{(s)}_{\text{word}}}$ whose cell (l, m) equals the number of regressive eye fixations moving from word l to word m. We extracted the lower-triangle entries (right-to-left saccades), flattened them, and concatenated across sentences to obtain one vector per participant of length $N_{\text{pairs}} = \sum_s n^{(s)}_{\text{pairs}}$. The same flattening procedure was applied to the attention vectors. We restricted the analysis to regressive saccades because (i) unidirectional LLM attention cannot, on geometric grounds, align with forward saccades, and (ii) regressive saccades index integration and re-reading processes (Gao et al., 2025). We acknowledge this captures only a subset of reading behavior (see Limitations).

Attention–saccade alignment. For each layer j, participant i, and condition, we computed the Pearson correlation r between the participant's regressive-saccade vector $V^{(i)}_{\text{sac}}$ and the model's attention vector on the same pair coordinates. At each layer we max-pooled attention across heads, yielding one value per pair; we took lower-triangle entries (right-to-left pairs) from the word-level matrices in the word condition and from chunk-pooled matrices in the phrase condition, using the same segmentation for attention and saccades, and concatenated across sentences. Absolute r values are numerically small because each entry is a single regressive-saccade count for one pair in one sentence; condition effects are best read as *relative* differences on the same data. For visualization (Figure 2), we additionally report r divided by a leave-one-out noise ceiling: the mean Pearson r between each participant's saccade vector and the average of the remaining $N-1$ participants' vectors (cf. Gao et al., 2025).

Statistical model. Linear Mixed-Effects (LME) models were fitted on the raw (unnormalized) r values. We included Model (Base vs. Fine-tuned) and Condition (Word, Phrase) as fixed effects and both Subject and Transformer Layer as random intercepts:

$$r \sim \text{Model} \times \text{Condition} + (1|\text{Subject}) + (1|\text{Layer}) \quad (1)$$

Partial correlation analysis. To isolate phrase boundaries from word-level attention, we upsampled chunk-pooled attention back to word-pair coordinates (each word pair inherits its chunk-pair value), placing word, phrase, and shuffled conditions in a common word-pair space. For each participant and layer we computed partial Pearson correlations $r(V_{\text{sac}}, V_{\text{phrase}} \mid V_{\text{word}})$ and $r(V_{\text{sac}}, V_{\text{shuf}} \mid V_{\text{word}})$ using the standard partial-correlation formula; shuffled boundaries were averaged over 50 random permutations per sentence.

2.7 Neural Encoding

We assessed model–brain alignment using encoding models that predict cortical activity from LLM attention features. To avoid circularity, we used a leave-one-subject-out (LOSO) procedure: for each held-out subject, the optimal transformer layer was selected from the remaining $N-1$ subjects, and the held-out encoding map at that layer was entered into a group-level spatial cluster permutation test (10,000 permutations, cluster-forming $p < 0.01$) (Maris and Oostenveld, 2007). We conducted (1) single-model encoding and (2) between-model comparisons (fine-tuned minus base). Clusters were labeled using the Desikan–Killiany atlas (Desikan et al., 2006).

Region-of-interest selection. Analyses were restricted *a priori* to a left-hemisphere language mask: the union of Desikan–Killiany parcels in left STG, middle temporal, inferior temporal, supramarginal, angular gyrus, and inferior frontal gyrus (pars opercularis, pars triangularis, pars orbitalis). This mask was fixed before any model analyses and reflects the left-lateralized language network in prior work (Fedorenko et al., 2011; Pallier et al., 2011); no right-hemisphere or non-language parcels were tested.

Representational similarity analysis (RSA). For each layer and condition, we computed RDMs from the model's attention patterns and from multivoxel STG activation patterns at the sentence level. The Spearman correlation between model and brain RDMs is the raw RSA measure, normalized by a leave-one-out lower-bound noise ceiling (Nili et al., 2014): for each subject the ceiling is the Spearman correlation between that subject's RDM and the average RDM of the remaining $N-1$ subjects. We fitted LME models with Condition and Model as fixed effects and random intercepts for Subject and Layer:

$$\text{RSA} \sim \text{Condition} \times \text{Model} + (1|\text{Subject}) + (1|\text{Layer}) \quad (2)$$

Per-condition follow-ups tested the effect of Model within each condition; p-values used the Satterthwaite approximation.

3 Results

3.1 Behavioral Alignment: Eye Movements

We aligned model attention with human reading by computing Pearson correlations between word-pair (or chunk-pair) attention and regressive-saccade counts on the same sentences (§ Methods; following Gao et al., 2025). For each participant and layer we obtained a raw correlation r; LME models were fitted on these raw values (Table 1). Figure 2 displays the same correlations normalized by a leave-one-out inter-subject noise ceiling for interpretability.

Table 1: Fixed effects from the LME model ($r \sim$ Model $\times$ Condition $+$ (1|Subject) $+$ (1|Layer)) fitted on raw Pearson r values. Reference levels: Base model, Word condition. Significance: $^*p < .05$, $^{***}p < .001$.

Predictor	β	p
Fine-tuned (vs. Base)	+0.004	<.001***
Phrase (vs. Word)	+0.030	<.001***
Fine-tuned × Phrase	+0.002	.024*

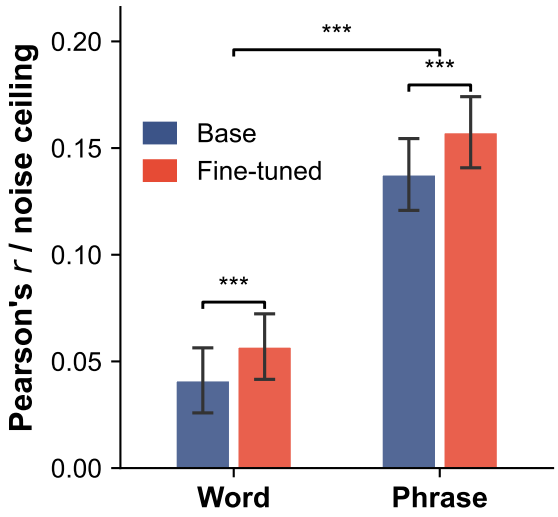


Figure 2: Eye-tracking–attention correlation by condition and model. Values are normalized by a leave-one-out noise ceiling computed from inter-subject regressive-saccade consistency. Bars show mean Pearson r / noise ceiling (averaged across layers); error bars: ± 1 SEM across subjects; brackets: per-condition fine-tuning contrasts.

Phrase superiority and fine-tuning. A robust main effect of Condition emerged ($F(1, 6316) = 6662.9$, $p < .001$): phrase-level features produced higher eye-tracking alignment than word-level features ($\beta = +0.030$, $p < .001$). Fine-tuning yielded a smaller but reliable improvement ($F(1, 6316) = 176.0$, $p < .001$; $\beta = +0.004$). The positive interaction ($\beta = +0.002$, $p = .024$) indicates a slightly larger fine-tuning benefit for the phrase condition. Raw r values are numerically small (word-level means ≈ 0.011–0.015; phrase-

Table 2: Partial correlation analysis controlling for word-level attention. Partial r: subject means across layers. Linguistic phrase boundaries yield positive partial correlations; shuffled boundaries yield negative values.

Model	Condition	Partial r	p
Base	Phrase	$+0.006$	.008**
	Shuffled	-0.008	$<.001$***
	Phrase − Shuffled	$+0.014$	$<.001$***
Fine-tuned	Phrase	$+0.008$	$<.001$***
	Shuffled	-0.008	$<.001$***
	Phrase − Shuffled	$+0.015$	$<.001$***

level ≈ 0.040–0.046); the phrase superiority effect is best read as a *relative* difference between conditions on the same data (cf. Gao et al., 2025).

Partial correlation: controlling for aggregation and lexical attention. The phrase superiority effect could partly reflect an aggregation artifact: pooling multiple word-level values into a single phrase-level value mechanically increases correlations regardless of linguistic structure. To isolate the contribution of *linguistic* phrase boundaries, we computed partial Pearson correlations $r(V_{\text{sac}}, V_{\text{phrase}} \mid V_{\text{word}})$ and $r(V_{\text{sac}}, V_{\text{shuf}} \mid V_{\text{word}})$ in a common word-pair space (§ Methods; Figure 3; Table 2). If the phrase advantage were solely an aggregation artifact, both phrase and shuffled boundaries should show comparable partial correlations once word-level attention is controlled.

Linguistic phrase boundaries showed a robust *positive* partial correlation ($r = +0.006$–$+0.008$, $p \leq .008$), while shuffled boundaries yielded a *negative* partial correlation ($r = -0.008$, $p < .001$). The Phrase−Shuffled gap was substantial ($\Delta = +0.014$–$+0.015$, $p < .001$) and replicated across both models. Fine-tuning further amplified this effect: the fine-tuning gain on phrase partial r ($\Delta = +0.002$, $p < .001$) exceeded the gain on shuffled partial r ($\Delta = +0.001$, $p < .001$), with a significant interaction ($p < .001$). As discussed in § Methods, partial correlations between correlated regressors systematically underestimate effect sizes, so the absolute values here should be interpreted as evidence that a real signal exists beyond word-level confounds, rather than as estimates of the total phrase contribution.

3.2 Neural Encoding: fMRI

We used LOSO cross-validated encoding analyses (§ Methods) to identify, within an a priori left-hemisphere language mask, brain regions where each model significantly predicts cortical activity, and to test where fine-tuning improves encoding strength.

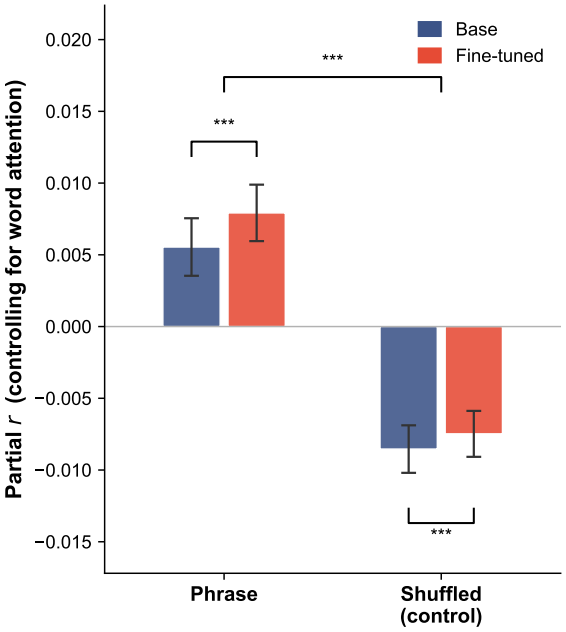


Figure 3: Partial correlation controlling for word-level attention. Each bar shows partial Pearson r between regressive-saccade counts and upsampled phrase-level (or shuffled) attention, with word-level attention partialed out. Error bars: ± 1 SEM across subjects; brackets: paired comparisons.

3.2.1 Single-Model Encoding

We first asked whether each model–condition combination produces encoding values significantly above zero, establishing that model attention predicts cortical activity.

Word and phrase conditions. Both models showed significant encoding of word and phrase representations in left STG (534 vertices, $p < .001$) and pars opercularis (101 vertices, $p < .01$) for all model $\times$ condition combinations (Figure 4, Panel A). LOSO consistently selected Layer 1 across all 50 subjects. The dominance of Layer 1 in single-model encoding most plausibly reflects shallow, input-driven features (e.g., positional or frequency-related regularities) that co-vary with STG activity regardless of fine-tuning; we return to this point in the Discussion.

Phrase − word residual. Neither model showed significant encoding for the phrase − word contrast, indicating that neither captures structure that is separable from word-level representations at the level

Table 3: Between-model encoding comparison (Fine-tuned > Base; cluster-forming $p < .01$, 10,000 permutations). n_{vtx} = cluster extent in vertices (effect size). Fine-tuning selectively enhanced encoding for phrase representations only. No significant clusters were observed for the word or phrase − word conditions.

Cond.	Region	n_{vtx}	p_{cluster}	LOSO Layer
Word	*No significant clusters*			
Phrase	STG	534	<.001***	L19 (48/50)
	Pars oper.	101	.009**	

Table 4: Per-condition LME effects of fine-tuning on STG representational alignment. β = unstandardized effect size (positive indicates greater alignment for the fine-tuned model). The phrase − word contrast was not significant ($\beta = +0.0003$, $p = .813$). The word effect ($p = .044$) does not survive Bonferroni correction for three comparisons ($\alpha_{\text{corrected}} = .017$).

Condition	β (Fine − Base)	p
Phrase	+0.0036	.004**
Word	+0.0033	.044*

our analysis can detect. This null constrains the interpretation of the phrase advantage to reflect representational *granularity* (the scale at which features are aggregated) rather than a distinct compositional mechanism.

3.2.2 Between-Model Comparison: Effect of Fine-Tuning

To identify regions where fine-tuning improved encoding, we compared the fine-tuned model against the base model within each condition (Table 3; Figure 4, Panel B).

Fine-tuning selectively enhanced encoding for *phrase* representations in left STG (534 vertices, $p < .001$) and pars opercularis (101 vertices, $p = .009$), with LOSO selecting Layer 19 for 48/50 subjects. No significant fine-tuning effect emerged for word representations or for the phrase − word residual. This combination of effects—condition-specific (phrase only), layer-specific (Layer 19), and anatomically localized—is what we would expect if fine-tuning sharpened intermediate-layer attention toward phrase-level structure that is also tracked by STG and pars opercularis. We do not interpret it as evidence for new compositional computation, given the phrase − word null (Discussion).

3.3 Representational Similarity Analysis

To complement the encoding analyses, we evaluated whether fine-tuning altered the geometric alignment between model attention and human STG activity using RSA (§ Methods). Results are summarized in Table 4 and Figure 5.

The LME model revealed significant main effects of Condition ($F(2) = 675.15$, $p < .001$) and Model ($F(1) = 7.59$, $p = .006$), with no significant interaction ($p = .218$). Per-condition follow-ups (Table 4) showed that fine-tuning significantly increased STG alignment for phrase representations ($\beta = +0.0036$, $p = .004$). A nominally significant word effect ($\beta = +0.0033$, $p = .044$) does not survive Bonferroni correction across the three conditions ($\alpha_{\text{corrected}} = .017$). No effect was observed for the phrase − word residual ($p = .813$).

The partial dissociation between RSA and encoding results is expected given their different measurement targets: encoding assesses vertex-wise predictive power, whereas RSA evaluates global geometric similarity. The most conservative reading is that fine-tuning primarily strengthens phrase-level correspondence, with possible but unconfirmed extension to word-level geometry (Figure 5).

4 Discussion

We asked whether phrase-level aggregation of LLM attention provides a closer correspondence to human reading behavior and language cortex than word-level attention, and whether phrase-segmentation fine-tuning amplifies that correspondence. Three converging lines of evidence support this view, with the caveat that "phrase" here refers to shallow chunks rather than full hierarchical constituents.

Phrase boundaries carry signal beyond aggregation and lexical attention. The partial correlation analysis shows that linguistic chunk boundaries explain unique variance in regressive eye saccades beyond word-level lexical attention and multi-word aggregation. Shuffled boundaries, which carry the same aggregation structure but disrupt linguistic groupings, yield negative partial correlations once word-level attention is controlled—directly addressing the concern that the phrase superiority effect is "merely aggregation smoothing." Absolute partial r values are small; partial correlations between correlated regressors systematically underestimate effects, so the relevant quantity is the small but reliable gap between Phrase and Shuffled. This pattern is consistent with recent independent

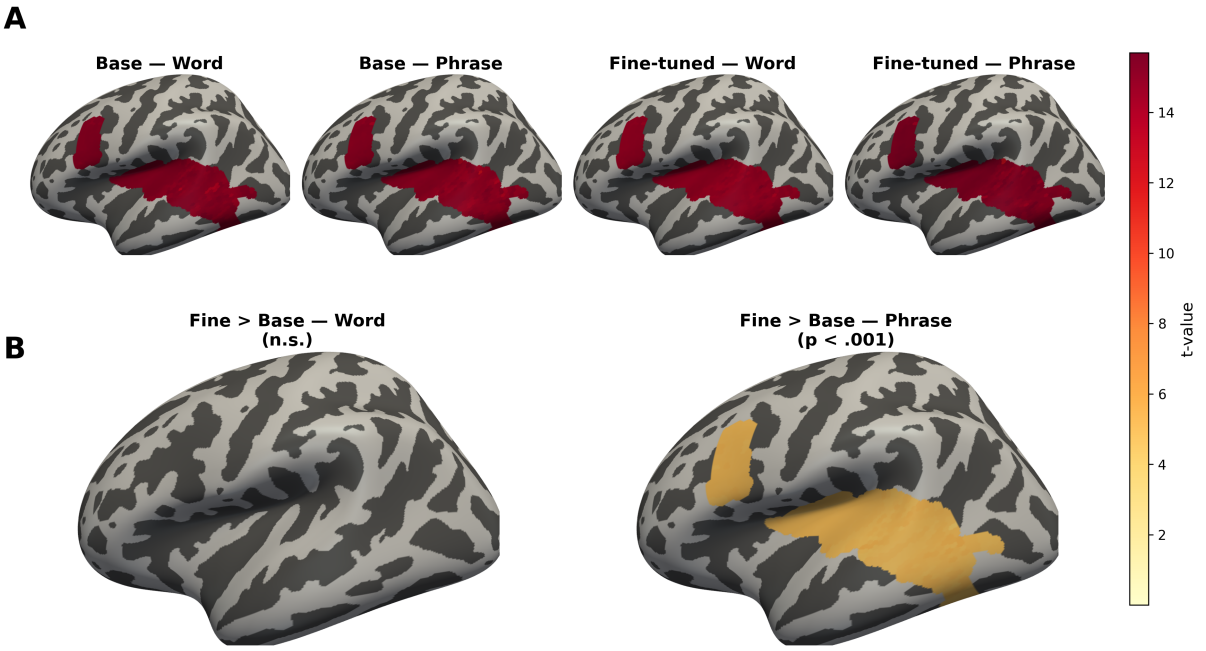


Figure 4: Neural encoding results. **(A)** Single-model encoding: both models predict cortical activity in left STG and pars opercularis (LOSO Layer 1; cluster-forming $p < .01$). **(B)** Between-model comparison: fine-tuning selectively enhances phrase encoding ($p < .001$); no word enhancement. LOSO layer distributions in Appendix Figure 6.

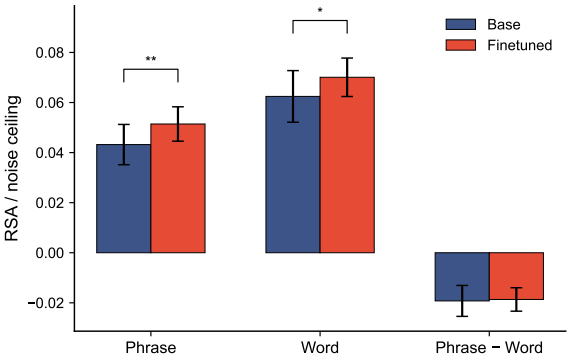


Figure 5: RSA summary: mean representational alignment (Spearman ρ / noise ceiling) between LLM attention and human STG activity. Fine-tuning significantly improves phrase alignment ($p = .004$); the word effect ($p = .044$) does not survive correction. Layer-wise profiles in Appendix Figure 7.

evidence that LLMs maintain latent tree-structured sentence representations that can be probed behaviorally (Liu et al., 2026), although our data do not, on their own, establish that the chunk-level signal we extract is hierarchical in that sense.

Fine-tuning selectively amplifies phrase alignment. Fine-tuning produces phrase-specific effects in both domains. Behaviorally, the gain on phrase boundaries exceeds the gain on shuffled boundaries (phrase-specific interaction). Neurally, the encoding enhancement is condition-specific (phrase only), layer-specific (Layer 19 in 48/50 subjects), and anatomically localized (left STG and pars opercularis within the a priori language mask); RSA confirms a phrase-level geometric improvement that survives Bonferroni correction. This combination (condition $\times$ layer $\times$ region) is what we would expect if fine-tuning sharpened intermediate-layer attention toward chunk-level structure also tracked by left language cortex. The divergence between Layer 1 (best single-model encoder) and Layer 19 (locus of fine-tuning improvement) is informative: Layer 1 attention likely captures shallow, input-driven regularities, whereas the Layer 19 gain reflects more abstract, context-sensitive representations (Tenney et al., 2019) that fine-tuning amplifies.

Constraint and scope. The absence of significant phrase $-$ word encoding or RSA effects in both models suggests that the phrase advantage reflects representational *granularity*, the scale at which features are extracted and pooled, rather than a separable compositional operation (Manning et al., 2020). Our results are best read as evidence about *how to aggregate* attention to align with human data, not as a claim that the LLM has acquired explicit compositional structure. Relative to prior work showing that scaling rather than

generic instruction tuning drives model–brain alignment (Gao et al., 2025; Pasquiou et al., 2022; Antonello et al., 2024), our results indicate that a *targeted, structure-relevant* fine-tuning objective can yield condition- and region-specific improvements not visible at the word level (Gwilliams et al., 2024; Tuckute et al., 2024). Caveats are discussed in the Limitations.

5 Conclusion

Within a single LLM and a single English reading dataset, aggregating attention at the shallow phrase (chunk) level yields closer alignment to eye movements and left language cortex than word-level attention. Phrase-segmentation fine-tuning selectively amplifies this in left STG and pars opercularis at intermediate layers; the persistent phrase − word null suggests representational granularity rather than separable composition.

Limitations

Our study has several limitations. *Single model:* all analyses use a single base LLM (Meta-Llama-3.1-8B) and its phrase-segmentation fine-tuned variant; demonstrating that the phrase superiority effect and its fine-tuning amplification replicate across model families and scales (e.g., GPT-2, Llama 7B/13B/70B) would substantially strengthen the generality of our conclusions. *"Phrase" = shallow chunk:* throughout, "phrase" refers to a shallow surface chunk produced by a rule-based segmentation pipeline (Appendix A), not a full hierarchical constituent; alternative segmentation frameworks (full constituency/dependency parses, or different chunking conventions such as CoNLL-2000) may yield different alignment patterns. *Eye-tracking measure:* the behavioral analysis uses only word-pair regressive-saccade counts, following Gao et al. (2025), because they share the lower-triangular word-pair geometry of right-to-left LLM attention. Regressive saccades are a relatively small fraction of the eye-tracking data and can be influenced by low-level factors that we did not control for (e.g., word length and frequency at the source and target word, fixation landing position, and visuospatial anchoring on physical layout); extension to forward-saccade or fixation-duration measures with explicit confound controls would strengthen the behavioral conclusion. *Numerically small correlations:* absolute Pearson r values are small and should be read as relative differences between conditions on the same data, not as estimates of absolute explained variance. *Partial correlation framework:* the partial-correlation upsamples phrase-level (and shuffled) attention back to word-pair space and may introduce residual correlation structure; independent replication with matched stimuli would further consolidate the conclusion. *fMRI scope:* encoding analyses focus on an a priori left-hemisphere language mask and a particular fixation-to-BOLD alignment procedure; broader cortical regions and alternative alignment assumptions could refine the anatomical conclusions. *Attention vs. hidden states:* the model side of all analyses is attention weights, not hidden states, to match the word-pair geometry of regressive saccades; the selection of Layer 1 in single-model encoding may therefore reflect shallow attention features (e.g., positional or frequency-related patterns) rather than deep linguistic computation. *Correlational evidence:* stronger model–brain correspondence does not establish mechanistic equivalence; causal links will require targeted interventions (e.g., attention-head ablation) and controlled stimuli.

References

Steven P Abney. 1991. Parsing by chunks. In *Principle-based parsing: Computation and Psycholinguistics*, pages 257–278. Springer.

Richard Antonello, Aditya Vaidya, and Alexander G. Huth. 2024. Scaling laws for language encoding models in fmri. *Preprint*, arXiv:2305.11863.

Kathryn Bock and Carol A Miller. 1991. Broken agreement. *Cognitive Psychology*, 23(1):45–93.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 632–642.

Zhenguang Cai, Xufeng Duan, David Haslett, Shuqi Wang, and Martin Pickering. 2024. Do large language models resemble humans in language use? In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 37–56.

Charlotte Caucheteux and Jean-Rémi King. 2022. Brains and algorithms partially converge in natural language processing. *Communications Biology*, 5(1):134.

Rahul S Desikan, Florent Ségonne, Bruce Fischl, Brian T Quinn, Bradford C Dickerson, Deborah Blacker, Randy L Buckner, Anders M Dale, R Paul Maguire, Bradley T Hyman, and 1 others. 2006. An

automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *Neuroimage*, 31(3):968–980.

Xufeng Duan, Bei Xiao, Xuemei Tang, and Zhenguang G Cai. 2024. Hlb: Benchmarking llms' humanlikeness in language use. *arXiv preprint arXiv:2409.15890*.

Evelina Fedorenko, Michael K. Behr, and Nancy Kanwisher. 2011. Functional specificity for high-level linguistic processing in the human brain. *Proceedings of the National Academy of Sciences*, 108(39):16428–16433. Epub 2011 Sep 1.

Evelina Fedorenko, Terri L. Scott, Peter Brunner, William G. Coon, Brianna Pritchett, Gerwin Schalk, and Nancy Kanwisher. 2016. Neural correlate of the construction of sentence meaning. *Proceedings of the National Academy of Sciences*, 113(41):E6256–E6262.

D. Jake Follmer, Szu-Han Fang, Roy B. Clariana, Bonnie J. F. Meyer, and Ping Li. 2018. What predicts adult readers' understanding of STEM texts? *Reading and Writing*, 31:185–214.

Changjiang Gao, Zhengwu Ma, Jiajun Chen, Ping Li, Shujian Huang, and Jixing Li. 2025. Increasing alignment of large language models with language processing in the human brain. *Nature computational science*, 5(11):1080–1090.

Merrill F Garrett. 1975. The analysis of sentence production. *Psychology of Learning and Motivation*, 9:133–177.

Edward Gibson. 1998. Linguistic complexity: locality of syntactic dependencies. *Cognition*, 68(1):1–76.

Adele E Goldberg. 2003. Constructions: A new theoretical approach to language. *Trends in cognitive sciences*, 7(5):219–224.

Ariel Goldstein, Zaid Zada, Eliav Buchnik, Mariano Schain, Amy Price, Bobbi Aubrey, Samuel A. Nastase, Amir Feder, Dotan Emanuel, Alon Cohen, Aren Jansen, Harshvardhan Gazula, Gina Choe, Aditi Rao, Catherine Kim, Colton Casto, Lora Fanda, Werner Doyle, Daniel Friedman, and 13 others. 2022. Shared computational principles for language processing in humans and deep language models. *Nature Neuroscience*, 25(3):369–380.

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

Laura Gwilliams, Alec Marantz, David Poeppel, and Jean-Remi King. 2024. Hierarchical dynamic coding coordinates speech comprehension in the brain. *bioRxiv*.

Chun-Ting Hsu, Roy Clariana, Benjamin Schloss, and Ping Li. 2019. Neurocognitive signatures of naturalistic reading of scientific texts: a fixation-related fMRI study. *Scientific Reports*, 9(1):1–16.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Jennifer Hu, Idan A. Blank, and Evelina Fedorenko. 2023. Precision fmri reveals that the language-selective network supports both phrase-structure building and lexical access during language production. *Cerebral Cortex*, 33(8):4384–4404. Published online 2022 Sep 20.

Ping Li and Roy Clariana. 2019. Reading comprehension in L1 and L2: An integrative approach. *Journal of Neurolinguistics*, 50:94–105.

Wei Liu, Ming Xiang, and Nai Ding. 2026. Active use of latent tree-structured sentence representation in humans and large language models. *Nature Human Behaviour*, 10(2):303–316.

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.

Eric Maris and Robert Oostenveld. 2007. Nonparametric statistical testing of eeg-and meg-data. *Journal of neuroscience methods*, 164(1):177–190.

Hamed Nili, Cai Wingfield, Alexander Walther, Li Su, William Marslen-Wilson, and Nikolaus Kriegeskorte. 2014. A toolbox for representational similarity analysis. *PLoS computational biology*, 10(4):e1003553.

Christophe Pallier, Anne-Dominique Devauchelle, and Stanislas Dehaene. 2011. Cortical representation of the constituent structure of sentences. *Proceedings of the National Academy of Sciences*, 108(6):2522–2527.

Alexandre Pasquiou, Yair Lakretz, John Hale, Bertrand Thirion, and Christophe Pallier. 2022. Neural language models are not born equal to fit brain data, but training helps. *Preprint*, arXiv:2207.03380.

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

Keith Rayner and Susan A. Duffy. 1986. Lexical complexity and fixation times in reading: Effects of word frequency, verb complexity, and lexical ambiguity. *Memory & Cognition*, 14(3):191–201.

Martin Schrimpf, Idan Asher Blank, Greta Tuckute, Carina Kauf, Eghbal A. Hosseini, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2021.

The neural architecture of language: Integrative modeling converges on predictive processing. *Proceedings of the National Academy of Sciences*, 118(45):e2105646118.

Nathaniel J. Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601. Association for Computational Linguistics.

Greta Tuckute, Nancy Kanwisher, and Evelina Fedorenko. 2024. Language in brains, minds, and machines. *Annual Review of Neuroscience*, 47.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2019. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Hanlin Wu, Xufeng Duan, and Zhenguang Cai. 2025. Distinct social-linguistic processing between humans and large audio-language models: Evidence from model-brain alignment. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 135–143. NAACL 2025.

A Fine-Tuning Details

A.1 Training Hyperparameters

We fine-tuned Meta-Llama-3.1-8B using the Unsloth FastLanguageModel framework with LoRA adapters applied to query, key, value, and output projection matrices ($r = 64$, $\alpha = 128$, dropout = 0.05). Gradient checkpointing was enabled for memory efficiency. Training used the TRL SFTTrainer with batch size 32 and gradient accumulation over 8 steps (effective batch size 256), AdamW optimizer with 8-bit quantization, learning rate 3×10^{-4} with cosine annealing, and 2 epochs with sequence packing. For inference, we used deterministic decoding (temperature = 0) with a custom stop token "]" and maximum new generation length of 64 tokens.

A.2 Phrase Segmentation Rules

The fine-tuning targets were generated by a deterministic rule-based pipeline applied to constituency parses produced. The rules are intentionally shallow and framework-neutral, producing what the NLP literature would describe as "chunks" rather than recursive constituents:

1. Preserve original word order—only insert pipe symbols (|) between phrases.
2. Keep the subject noun phrase intact as the first segment.
3. Isolate the main verb or predicate as a separate segment.
4. Group objects and complements (if any) into subsequent segments.
5. Separate adverbial modifiers (prepositional phrases, adverbs, etc.) into their own segments.
6. For embedded clauses, recursively apply the same rules to split them into subject-verb-object structure.
7. Treat coordinated elements (e.g., "bread and butter") as one phrase.
8. Enclose all output results with square brackets.

Quality of the segmentation on the ReadBrain stimuli. The ReadBrain stimuli contain technical STEM language for which automatic parsers can be unreliable, so we additionally manually inspected the rule-based output. Two co-authors independently judged whether the rule-based phrase boundaries on a random sample of 200 ReadBrain sentences were linguistically plausible (i.e., did not split a base NP, break a verb group, or place a boundary mid-PP). Boundary-by-boundary agreement was 92.7%, with disagreements concentrated on prepositional-phrase attachment ambiguities (e.g., locative vs. instrumental PPs). We acknowledge that this is not an external gold-standard evaluation; we did not use a pre-existing gold-parsed dataset because available chunking corpora (e.g., CoNLL-2000) do not match the genre and vocabulary of the STEM articles in ReadBrain. The fine-tuned LLM achieves 94.2% exact-string match against this rule-based gold on a held-out test set of 140 ReadBrain sentences, vs. 25.7% for the base model.

A.3 Example Segmentations

Simple sentence:
Input: The quick brown fox jumps over the lazy dog.
Output: [The quick brown fox | jumps | over the lazy dog | .]
Relative clause:
Input: The book that I read last night was fascinating.
Output: [The book | that I read | last night | was | fascinating | .]
Complex sentence:
Input: After the discussion, I will carefully conduct the experiment based on the results.
Output: [After the discussion | , | I | will | carefully | conduct | the experiment | based | on the results | .]

B Supplementary Figures

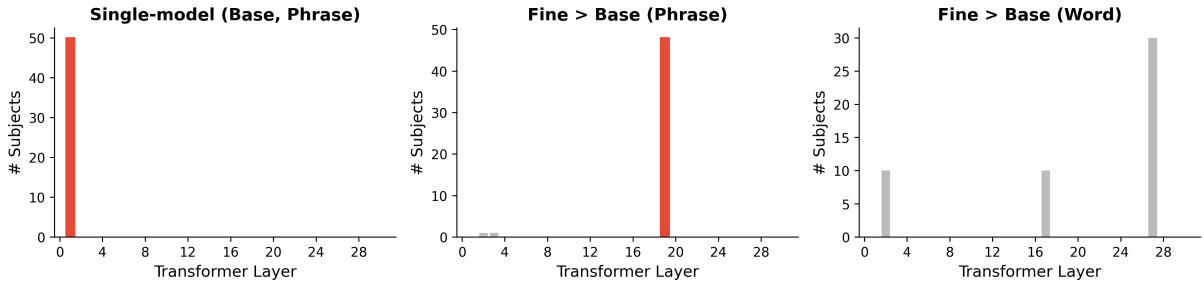


Figure 6: LOSO best-layer distribution for encoding analyses. The single-model phrase analysis and the between-model phrase comparison converge on a single dominant layer (Layer 1 and Layer 19, respectively; red bars), whereas the non-significant word comparison yields a dispersed distribution across Layers 2, 17, and 27 (grey bars), consistent with the layer-specificity of the phrase-level fine-tuning effect.

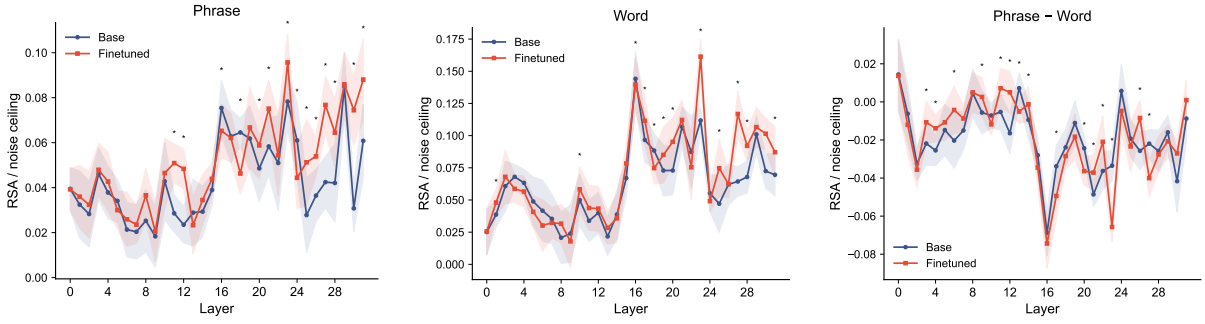


Figure 7: Layer-wise RSA profiles (Spearman ρ / noise ceiling) for phrase, word, and phrase−word conditions. Blue: base model; red: fine-tuned model. Shaded regions denote ± 1 SEM across subjects ($N = 50$). Stars indicate layers with significant paired-t difference ($p < .05$). Both models show rising RSA in middle-to-late layers; the fine-tuned model achieves higher alignment for the phrase condition.

Roles of Predictability and Acoustic Distance in Sound Discrimination via Contrastive Learning

Shuhao Zhang and Youngah Do
The University of Hong Kong
shuhaoz19@connect.hku.hk, youngah@hku.hk

Abstract

Research in sound discrimination demonstrates that listeners exhibit reduced sensitivity to acoustic differences between allophones, as opposed to phonemes. Previous studies indicates that highly predictable, complementary distribution of allophones contributes to this limited sensitivity by providing strong contextual cues. Building on these insights, this study investigates the role of predictability in sound discrimination within a supervised contrastive learning framework. Specifically, we examine how varying levels of predictability affect the ability to distinguish sounds and whether this influence is categorical or gradual. Additionally, we explore the interaction between acoustic distance and predictability, as well as how the presence of other contrasts within a language modulates this process. Our findings indicate that only full predictability leads to a significant decline in discrimination performance, demonstrating a categorical effect. This impairment can be alleviated as acoustic distance increases. Moreover, the presence of additional contrasts sharing the relevant acoustic dimension enhances discriminability, showing the importance of contextual contrasts in speech perception.

1 Introduction

When discriminating between different sounds, studies have shown that listeners perceive allophones within their native language as less distinct than phonemes, and this phenomenon emerges early in language development (Whalen et al., 1997; Pegg and Werker, 1997; Peperkamp et al., 2006; Boomershine et al., 2008; Seidl et al., 2009; Seidl and Cristia, 2012). For instance, in Boomershine et al. (2008), both Spanish-speaking and English-speaking listeners were asked to rate the similarity of sound pairs that differ in phonological status across the two languages (e.g., /d/ and /ð/) . In English, these are separate phonemes, whereas in Spanish, they are allophones in complementary environments. The results reveal that Spanish-speaking listeners rated these complementarily distributed sound pairs as more similar than English-speaking listeners did, indicating that listeners are less sensitive to allophonic differences than to phonemic distinctions, even when the acoustic distances between the sounds are comparable.

One possible explanation for this perceptual difference is that allophones generally occur in highly predictable, complementary environments, while phonemes can appear in overlapping contexts. Prior research suggests that contextual cues, especially complementary distributions, increase the predictability of sounds and thus attenuate individuals' sensitivity to acoustic differences (Peperkamp et al., 2003; Noguchi and Kam, 2018; Barrios et al., 2023). For example, Peperkamp et al. (2003) exposed participants to a continuum from [ʁ] to [χ], with stimuli presented in different distributions of occurrence (bimodal vs. unimodal distributions) and varying contextual conditions (overlapping vs. complementary conditions). The results of the post-exposure discrimination test suggest that, despite bimodal exposure, the high predictability of speech sounds from complementary contexts led to listeners' reduced sensitivity to the acoustic differences, compared to overlapping contexts. These studies indicate that increased predictability from complementary contexts reduces listeners' sensitivity to target sounds by enabling reliance on contextual cues (Peperkamp et al., 2003; Noguchi and Kam, 2018).

In natural languages, however, the predictability of sounds is not always defined in absolute terms. Instead, it is often gradient, with certain sounds more likely to occur in specific phonological environments than others. To systematically explore the role of predictability, beyond just the two extremes, the current study investigates how varying degrees of predictability influence discrimination performance along a continuum.

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 477–486
July 3-4, 2026 ©2026 Association for Computational Linguistics

In addition to the influence of complementary contexts on sound discrimination, acoustic distance within such distributions plays a key role in perception. Allophones tend to share similar acoustic properties (Peperkamp et al., 2006; Skoruppa et al., 2011) and generally exhibit smaller acoustic distances than phonemes (Seidl and Cristia, 2012). When two sounds are acoustically very distant, they are thus less likely to be perceived as allophones of the same phoneme, even within complementary contexts (Peperkamp et al., 2006).

Building upon these backgrounds and specifically extending the work of Peperkamp et al. (2003), the current study aims to compare discrimination performance of sounds across different levels of predictability, while systematically varying the acoustic distances within bimodal distributions. Our hypothesis is that exposure to the complementary distribution of two sounds (100% predictability) will lead to reduced sensitivity to acoustic differences; however, large acoustic distances may attenuate this effect even within complementary contexts. To capture the gradient nature of acoustic differences, we examine multiple sound pairs with graded acoustic distances, allowing us to observe how discrimination performance varies along a continuum of acoustic similarity.

Our study design thus addresses two main questions:

(a) Does full predictability (i.e., complementary context) lead to reduced sensitivity? If so, is the effect of predictability categorical (i.e., reflecting phoneme-like versus allophone-like perception), or does it change gradually along with predictability?

(b) How does the influence of contextual cues on sound discrimination change across varying levels of acoustic distances?

To investigate these questions, we employ a supervised contrastive learning framework (Khosla et al., 2020). The model is trained to distinguish between different CV syllables for each condition over 200 epochs, simulating human perception of sound pairs. The trained encoder is then used to evaluate the discrimination performance on a test set. By analyzing the latent feature representations using the silhouette score and the distance distribution overlap (DDO) score, we quantify how well the two categories in the test set are separated in the feature space.

2 Experiment 1

Experiment 1 examines how predictability influences the discrimination of speech sounds of different acoustic distance. Within a contrastive learning framework, sound pairs are presented with varying acoustic similarity, represented by acoustic vectors, within different contextual conditions to modulate predictability. Discrimination performance is assessed by evaluating the silhouette scores and DDO scores of the sounds embedded within the same context, providing insights into how the two factors interact to influence perceptual separation.

2.1 Method

2.1.1 Dataset

The stimuli used are CV syllables, consisting of eleven fricatives differing primarily in their center of gravity (CoG). These form ten pairs of sounds, with a shared anchor sound (labeled as /s_0/) and a variable second sound (/s_1/ to /s_{10}/, referred to as /s_x/). The ten pairs are labeled as pair 1 through pair 10.

The mean CoG of /s_0/ is 9000 Hz. For each /s_x/, the mean CoG decreases in steps of 500 Hz, starting from 8000 Hz for /s_1/ down to 3500 Hz for /s_{10}/. The distribution of each sound along the CoG dimension follows a truncated Gaussian distribution spanning $[\text{mean} - 2 \times \text{sd}, \text{mean} + 2 \times \text{sd}]$, with a standard deviation $\text{sd} = 500$ Hz. For instance, /s_1/ ranges from 7000 Hz to 9000 Hz. Consequently, in the design, the two sounds in pair 1 and pair 2 have overlapping values along the CoG dimension, respectively. The truncated gaussian distribution is selected to maintain control over the scope while approximating realistic distributional variability.

In the training set, we manipulate the distribution of target sounds to create a gradient of predictability levels, including 0%, 25%, 50%, 75%, and 100%. This is achieved by varying the proportion of complementary contexts, using eight vowels (/i/, /ɪ/, /e/, /ɛ/, /u/, /ʊ/, /o/, /ɔ/) as contexts to form CV syllables. For each pair of sounds (/s_0/ and /s_x/), predictability is determined by the proportion of stimuli with complementary contexts. For example, in the 75% predictability condition, vowels /i, ɪ, e/ are only preceded by /s_0/, while /u, ʊ, o/ only follow /s_x/. Vowels /ɛ, ɔ/ are shared between the two fricatives. The number of tokens involving each vowel is equal, resulting in stimuli with complementary contexts constituting 75% of all stimuli

in this condition.

In the test set, both sounds in each pair are combined with the following vowel /ɑ/. All nine vowels (eight in the training set and one in the test set) only differ in their first three formant frequencies (F1 to F3). The values are based on measurements from 48 female American English speakers by Hillenbrand et al. (1994).

Thus, this creates 50 conditions in total, varying in predictability (five levels ranging from 0% to 100%) and acoustic distance (ten pairs). Each condition has 1,000 tokens per vowel for the training set, totaling 8,000 tokens per condition.

2.1.2 Input manipulation

Stimuli are represented as acoustic vectors: each sound as a 16-dimensional feature vector, with pairs of sounds combined into a 2×16 vector representing a CV syllable. The 16 dimensions of each sound include: center of gravity, frication duration, standard deviation, skewness, kurtosis, frication intensity, total duration, vocalic duration, vocalic intensity, fundamental frequency (F0), and frequencies and bandwidths of the first three formants (F1, F2, F3). The structure is shown in Table 1.

For fricatives, the last nine features have a zero value, while vowels are assigned zero values for the first six features. Most features follow a truncated Gaussian distribution ranging between $[\text{mean} - 2 \times \text{sd}, \text{mean} + 2 \times \text{sd}]$, with a standard deviation set to 5% of the mean value for each (except for CoG, whose standard deviation is 500 Hz). Total duration and vocalic duration are constant. The mean values of all components, other than the formant frequencies and CoG values, are assigned to be consistent across different syllables, with reference to relevant phonetic studies (e.g., de Cheveigné, 1999; Jongman et al., 2000; McMurray and Jongman, 2011; Li and Gu, 2015). Sample vector values for a token of /s_0i/ are shown in Table 1.

2.1.3 Model structure and loss function

The core encoder is a fully connected neural network comprising a flattening layer, followed by two linear layers with ReLU activations, which compresses input vectors into a four-dimensional feature space.

The training optimization employs a supervised contrastive loss based on normalized temperature-scaled similarity. A typical function of supervised contrastive learning is illustrated in Equation 1 (Khosla et al., 2020).

$$L = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(\boldsymbol{z_i} \cdot \boldsymbol{z_p}/\tau\right)}{\sum_{a \in A(i)} \exp\left(\boldsymbol{z_i} \cdot \boldsymbol{z_a}/\tau\right)} \tag{1}$$

In this formulation, i denotes a specific token in the batch I, and $\boldsymbol{z_i}$ is its corresponding feature vector. The symbol $\cdot$ represents the inner (dot) product. The set $A(i)$ includes all samples in the batch I except for the token i, while $P(i)$ comprises all positive samples for i within $A(i)$, with $|P(i)|$ indicating its cardinality. τ is a scalar temperature parameter.

In our implementation, feature vectors are first normalized to unit length before similarity computation. The similarity matrix is then derived for all pairs within the batch. To improve numerical stability, the maximum similarity value in each row, corresponding to the similarity of the token i with itself, is subtracted from all entries in that row. A mask tensor is employed to distinguish positive pairs from negative pairs, while excluding self-similarities to prevent trivial solutions. To prevent division by zero, $|P(i)|$ is set to 1 when no positive samples are available for token i in this batch. The overall loss for the batch is calculated by averaging the individual losses of each sample. The loss function used in this study is shown in Equation 2.

$$\begin{aligned} Loss &= \frac{1}{N} \sum_{i \in I} \frac{-1}{D_i} \sum_{p \in P(i)} \log \frac{\exp S_{i,p}}{\sum_{a \in A(i)} \exp S_{i,a}} \\ S_{i,j} &= \text{sim}(\boldsymbol{z_i}, \boldsymbol{z_j})/\tau - \text{sim}(\boldsymbol{z_i}, \boldsymbol{z_i})/\tau \\ D_i &= \begin{cases} 1, & \text{if } |P(i)| = 0 \\ |P(i)|, & \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

In Equation 2, N is the number of samples in the batch I. The function $\text{sim}(\boldsymbol{z_i}, \boldsymbol{z_j})$ calculates the cosine similarity between the vectors. The temperature parameter τ is set to 0.07 in this study. This loss function promotes embeddings of samples from the same class to cluster together, while pushing apart embeddings from different classes, aligning with recent contrastive learning methodologies (e.g., Chen et al., 2020; Khosla et al., 2020; Gao et al., 2022; Liao et al., 2023).

No.	Element	Explanation	Distribution	/s$_0$/	/i/
1	cog	Center of gravity	Gaussian	8847.59	0.00
2	fri_dur	Frication duration	Gaussian	178.82	0.00
3	sta_dev	Standard deviation	Gaussian	180.72	0.00
4	skewness	Skewness	Gaussian	0.51	0.00
5	kurtosis	Kurtosis	Gaussian	0.92	0.00
6	fri_int	Frication intensity	Gaussian	61.39	0.00
7	tot_dur	Total duration	Fixed	200.00	400.00
8	voc_dur	Vocalic duration	Fixed	0.00	400.00
9	voc_int	Vocalic intensity	Gaussian	0.00	79.72
10	f0	Fundamental frequency	Gaussian	0.00	190.57
11	b3	Bandwidth of F3	Gaussian	0.00	169.54
12	b2	Bandwidth of F2	Gaussian	0.00	114.70
13	b1	Bandwidth of F1	Gaussian	0.00	86.99
14	f3	Frequency of F3	Gaussian	0.00	3335.26
15	f2	Frequency of F2	Gaussian	0.00	2911.90
16	f1	Frequency of F1	Gaussian	0.00	407.04

Table 1: Vector components and sample vector values for a token of /s$_0$i/

2.1.4 Training and evaluation

The model is trained for 200 epochs using batches of size 32, with a fixed learning rate of 0.001. During training, the model of each condition learns distinguish between different CV syllables, under the contrastive loss described earlier. After training, the encoder generates the four-dimensional feature embeddings for test data. Discrimination performance of the model is quantified using two metrics: the silhouette score and the DDO score. The silhouette score measures how well the two classes are separated in the embedding space, with higher values indicating greater separation and, consequently, better discrimination. The DDO score quantifies the overlap between the probability density functions of all pairwise within-class and between-class distances, with lower scores signifying improved discrimination.

To mitigate the effects of randomness, each condition is repeated ten times with independently initialized models, providing more robust estimates of performance for subsequent statistical analysis.

2.1.5 Analysis

For each step, a linear regression model is utilized to examine the effect of predictability on the response variable, the silhouette score or the DDO scores. Following this, post-hoc pairwise comparisons using Tukey's Honestly Significant Difference (HSD) test are performed to assess differences in the scores across various levels of predictability.

2.2 Results

The silhouette scores for each condition in Experiment 1 are displayed in Figure 1. The y-axis depicts the silhouette score, while the x-axis indicates the step of the sound being compared (i.e., /s$_x$/).

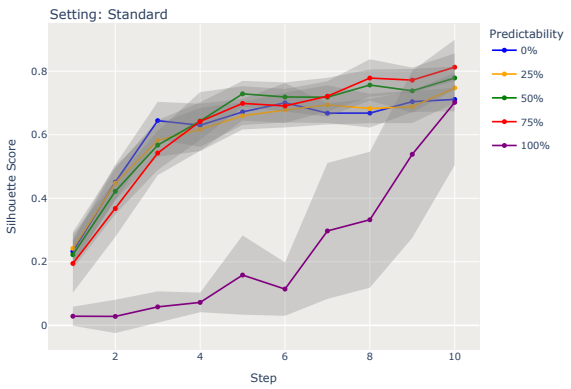


Figure 1: Silhouette scores for each condition in Experiment 1

Regarding the role of predictability, the results show that 100% predictability significantly leads to worse performance of sound discrimination compared to all other predictability levels at steps 1 to 8 (all p values < .0001). In contrast, the differences among the other predictability levels are not statistically significant (all p values > .05) across all acoustic distance conditions (i.e., steps).

There is a consistent trend that larger acoustic distances tend to correspond to higher silhouette score. When examining the interaction between

predictability and acoustic distance, it is notable that the influence of 100% predictability appears to diminish as the acoustic distance increases. Specifically, in conditions with large acoustic distances, there is limited or no significant difference in silhouette scores between the 100% predictability condition and other predictability levels. For example, at step 9, the difference in silhouette scores between the 100% and 25% predictability levels is not statistically significant (p = .077), and at step 10, no significant difference is observed between the 100% predictability condition and the other levels (all p values > .05). This suggests that salient acoustic distance can attenuate the influence of predictability on sound discrimination.

The DDO scores for each condition are shown in Figure 2. These scores provide similar information to the silhouette scores. However, even at step 10, 100% predictability does not achieve the same performance as other predictability levels (all p values < .01), although the differences are substantially reduced.

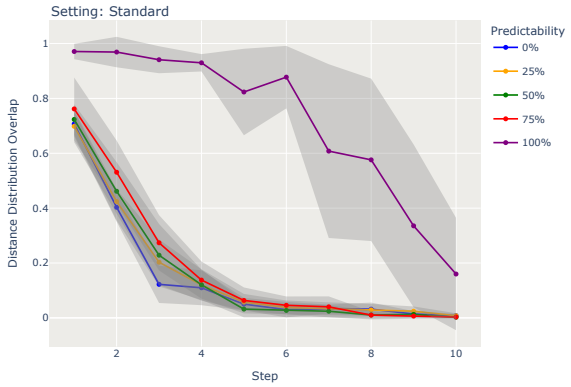


Figure 2: Distance distribution overlap (DDO) scores for each condition in Experiment 1

Overall, these findings indicate that predictability from contextual cues can influence sound perception, with full predictability reducing the sensitivity to the acoustic differences. The effect of predictability appears to be categorical, affecting performance only under complete predictability conditions. Meanwhile, acoustic distance gradually enhances discrimination performance, ultimately offsetting the influence from full predictability when the acoustic distance is sufficiently large.

3 Experiment 2

In Experiment 1, we have determined whether the effect of predictability from contextual cues is categorical or gradual and how this influence varies across different acoustic distances, focusing on a single contrast. However, in natural languages, contrasts are rarely learned in isolation. Instead, acoustic distinctions from other contrasts may also affect the perception of a new pair of sounds. To explore this, we conduct additional series of experiments to examine how acoustic distinctions from other sound contrasts impact the discrimination of the target sounds.

3.1 Method

Building on Experiment 1, we introduce an additional pair of sounds with overlapping contexts in training, where the predictability for this pair is set to 0%. For comparison, we train three sets of models, each incorporating a new sound pair with distinct levels in the CoG difference:

(1) A pair showing no difference in CoG, labeled as /z$_x$/ and /ts$_x$/, with their CoG values matching the second sound /s$_x$/ in the target sound pair for each condition.

(2) A pair exhibiting a fixed larger difference in CoG, labeled as /z$_0$/ and /z$_8$/, as their CoG values resemble those of /s$_0$/ and /s$_8$/.

(3) A pair with a fixed smaller difference in CoG, labeled as /z$_0$/ and /z$_5$/, as their CoG values are similar to those of /s$_0$/ and /s$_5$/.

Among them, /z$_0$/ and /z$_x$/ can be considered voicing counterparts to the target sounds /s$_0$/ and /s$_x$/, respectively, while /ts$_x$/ serves as an affricate counterpart of /s$_x$/. For simplicity, the three settings of models are designated as 'Unrelated', 'Long-Distance', and 'Short-Distance', respectively. The setting from Experiment 1 is referred to as 'Standard'. In Experiment 2, the training set is expanded by doubling the total number of tokens with an additional contrast, resulting in 16,000 tokens in each condition.

Regarding the input vectors, voicing differences are achieved by changing the vocalic duration of the consonant, whereas affricate differences are manipulated by altering the frication duration of the consonant. Specifically, the vocalic duration for voiceless fricatives is set to 0 ms, while for voiced counterparts it is fixed at 200 ms. Frication duration is 174 ms for fricatives, while it is 96 ms for affricates, both sampled from truncated gaussian distributions. The values for each component are

assigned based on established phonetic research (Jongman et al., 2000; Lee, 2011; Li and Gu, 2015). Other components are manipulated similarly to the target sounds in Experiment 1. The model structure, loss function, and training and evaluation procedures remain consistent throughout, except that only the silhouette scores are retained for more concise comparisons.

The same analysis pipeline used in Experiment 1 is applied to the results from each of the three model settings. Additionally, we compare the outcomes across the four settings based on their predictability, especially 100% predictability. For each step within each predictability level, we utilize a linear regression model to examine the effect of different experimental settings on the silhouette scores. Pairwise comparisons are conducted to assess the differences between various settings.

3.2 Results

The silhouette scores for each condition in the three settings of Experiment 2 are presented in Figure 3.

The results across all three settings suggest that 100% predictability tends to have a distinct impact compared to other predictability levels. In the 'Unrelated' setting, 100% predictability is significantly different from the other predictability levels from steps 1 to 9 with lower silhouette scores (except for the comparison with 75% predictability at step 1, where $p = 1$), indicating poorer discrimination. In the 'Long-Distance' setting, this significance is observed from steps 1 to 7 (except for the comparison with 75% predictability at step 6, where $p = .0824$), while it spans from steps 1 to 5 for the 'Short-Distance' setting. Aside from 100% predictability, other predictability levels show little or no difference at each step for all three settings.

In general, settings with another sound pair exhibit similar trends to the 'Standard' setting. However, a key difference lies in the influence of the 100% predictability. Figure 4 shows the silhouette scores for 100% predictability across the four settings at each step. Throughout all steps, the 'Standard' setting shows no significant difference from the 'Unrelated' setting. The 'Short-Distance' setting demonstrates significantly better discrimination than the 'Long-Distance' setting only at steps 3 ($p = .0002$) and 4 ($p = .0157$), with no other significant pairwise differences observed.

From steps 1 to 8, 'Standard' and 'Unrelated' settings generally show lower silhouette scores compared to the 'Long-Distance' and 'Short-Distance'

Figure 3: Silhouette scores for each condition in three model settings of Experiment 2

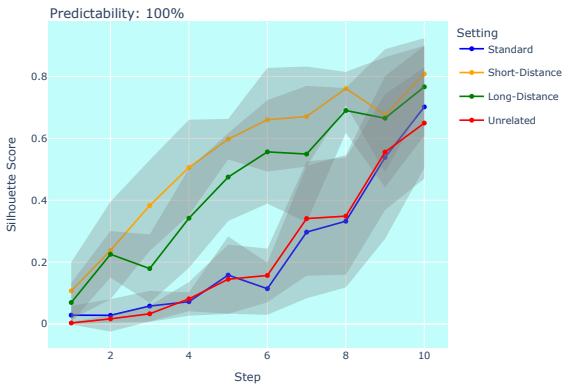


Figure 4: Silhouette scores for 100% predictability in different model settings

settings, with limited exceptions. However, this difference diminishes, when acoustic distance reaches steps 9 and 10. These findings suggest that another contrast sharing acoustic features in the target dimension facilitates the discrimination of a new sound pair, even when predictability of the target sounds is high and their acoustic distance is not very large. However, if this distinction cannot be learned from other sound contrasts, this facilitative effect is absent, resulting in discrimination performance comparable to that observed when learning in isolation. This underscores the essential role of shared acoustic features in supporting perceptual discrimination, particularly under conditions of high predictability and limited acoustic distance between target sounds.

4 Discussion

4.1 Role of predictability

The results reveal that discrimination performance under full predictability (100%) is significantly poorer compared to other levels of predictability, which do not differ significantly among themselves. This suggests that the influence of contextual cues on sound discrimination operates in a categorical manner: only when predictability reaches its maximum does perceptual sensitivity diminish noticeably. This phenomenon aligns with previous behavioral studies proposing that listeners treat sounds with fully predictable contexts (complementary distribution) as allophones, leading to reduced sensitivity to the acoustic differences, while phonemes remain encoded as contrastive (Peperkamp et al., 2003; Noguchi and Kam, 2018).

However, this categorical effect is not absolute. For example, in the 'Unrelated' setting, 75% predictability also shows significant differences from lower predictable levels at some steps (e.g., the pairwise comparison of 75% predictability and 50% predictability at step 4, with $p = .001$). This indicates that near-complete predictability might also influence discrimination, albeit with less consistency. A potential explanation for this general categorical manner is that the model's capacity or the task's simplicity may inherently limit the detection of subtle gradual changes in non-full predictability levels. Specifically, the model might tend to learn the contrast even when only a small amount of evidence is present showing that they are not fully predictable, thus the model separates the two sounds in its latent representations. As contrastive learning benefits more from longer training steps, later in training, the model is exposed to more negative pairs, thus more ready to learn the contrast even when overlapping contextual cues are limited (Chen et al., 2020).

Supporting this, analysis of early training phases for a closely paired contrast (first 10 epochs at step 3, under the 'Standard' setting) shows some gradual differences in the silhouette scores at lower predictability levels (see Figure 5). This suggests that the effect of predictability may have a gradual component that is hidden by the model's tendency to perform categorically at later stages. In other words, during early training, the model registers subtle gradations, but as training proceeds, it shifts to making a more binary, categorical distinction.

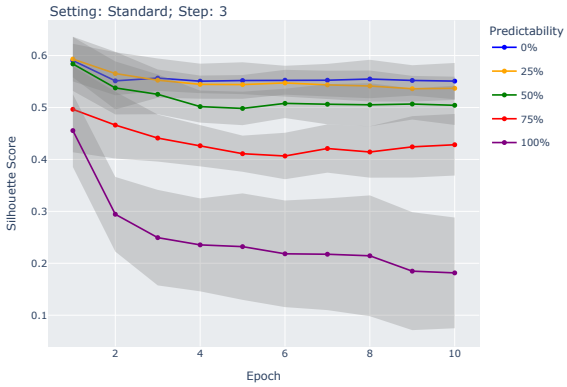


Figure 5: Silhouette scores across different epochs at step 3 in the 'Standard' setting

4.2 Role of acoustic distance and its interaction with predictability

Across all conditions, larger acoustic distances are associated with improved discrimination performance. Importantly, as acoustic distance increases, its influence begins to outweigh the effect of predictability, indicating that acoustic effects can mitigate context-driven perceptual attenuation.

Specifically, for the predictability condition that impacts discrimination the most (i.e., 100% predictability), the influence appears to change gradually along the acoustic distance. Rather than a sharp threshold, increasing acoustic differences lead to a progressive gain in the discrimination performance, constrained by the salience of acoustic cues. This implies that perceptual sensitivity is modulated by an incremental weighting of cue salience, where the effect of contextual predictability can be eventually overcome by salient acoustic cues, also sup-

porting that sound perception comprises utilization and interaction of multiple cues (Norris, 2003; Davis et al., 2005; Martin et al., 2013; Fourtassi and Dupoux, 2014; Frank et al., 2014).

4.3 Role of other sound contrasts

The presence of additional contrasts that share the same acoustic dimension enhances discrimination performance under full predictability. This suggests that phonemic or phonological information related to the relevant acoustic dimension does not operate in isolation but functions as a network of cues (Kuhl, 2000), reinforcing perceptual distinctions across contrasts. Moreover, when the acoustic distance of the additional sound pair is smaller, the discrimination accuracy will increase more.

This finding indicates that phonetic information from other contrasts can strengthen the perceptual salience of the target contrast. In other words, related phonetic cues from additional sounds can support and enhance the acoustic distinctions that are important for differentiating the target sounds (Maye et al., 2008; Finley and Badecker, 2009). This support helps to offset the decrease in perceptual sensitivity caused by high predictability, making subtle acoustic differences easier to detect. The effect is particularly prominent when the additional contrast involves a smaller acoustic difference. As a result, even slight acoustic distinctions become clearer because the influence of related contrasts facilitates the ability to discriminate, especially in environments where cues might otherwise be weakened by predictability.

Conversely, when the additional contrast exhibits no acoustic difference in the target dimension, indicating they are not related, it neither facilitates nor inhibits discrimination, highlighting that acoustic interactions are specific to contrasts sharing relevant phonetic features.

5 Limitations and conclusion

Our study utilizes a supervised contrastive learning framework to simulate the perception of sound pairs with different predictability levels and acoustic distances. The input consists of manipulated acoustic vectors with 16 features for each sound, derived from Gaussian distributions. Although this design is grounded in phonetic data and easy to control, it may not fully capture the variability and dynamic properties of natural speech. Additionally, the explicit supervision in the model limits its ability to reflect the complexity of natural language perception. Future work should consider more realistic inputs and improved model designs to enhance generalizability to real-world language perception. Apart from the influence of predictability and acoustic distance, the current study investigates the interactions with other contrasts, only considering one additional contrast for comparison. While the findings can be generalized to situations involving multiple sound pairs, incorporating a full phoneme inventory from a specific language would provide a more comprehensive understanding of how language-specific phonological systems influence perception.

Overall, the present study advances our understanding of how predictability and acoustic distance jointly influence sound discrimination via contrastive learning. The results demonstrate that predictability significantly impairs discrimination performance only when it reaches its maximum, suggesting a categorical influence of contextual cues on perceptual sensitivity. Furthermore, acoustic distance interacts incrementally with predictability, with larger acoustic distinctions counteracting the attenuating effect from full predictability on perception. The presence of related contrasts sharing phonetic features underscores the role of phonological networks in supporting perceptual robustness, especially under high predictability and limited acoustic distinction. These findings reinforce the notion that speech perception involves a complex interplay of multiple cues, where context, acoustic features, and phonological relationships among related sound pairs dynamically shape perceptual outcomes.

References

Shannon L Barrios, Joselyn M Rodriguez, and Taylor Anne Barriuso. 2023. The acquisition of L2 allophonic variants: The role of phonological distribution and lexical cues. *Second Language Research*, 39(3):899–924.

Amanda Boomershine, Kathleen Currie Hall, Elizabeth Hume, and Keith Johnson. 2008. The impact of allophony versus contrast on speech perception. In Peter Avery, B. Elan Dresher, and Keren Rice, editors, *Contrast in Phonology*, pages 145–172. Mouton de Gruyter.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv preprint*. ArXiv:2002.05709 [cs].

Matthew H. Davis, Ingrid S. Johnsrude, Alexis Hervais-Adelman, Karen Taylor, and Carolyn McGettigan. 2005. Lexical Information Drives Perceptual Learning of Distorted Speech: Evidence From the Comprehension of Noise-Vocoded Sentences. *Journal of Experimental Psychology: General*, 134(2):222–241.

Alain de Cheveigné. 1999. Formant bandwidth affects the identification of competing vowels. *ICPhS*, pages 2093–2096.

Sara Finley and William Badecker. 2009. Artificial language learning and feature-based generalization. *Journal of Memory and Language*, 61(3):423–437.

Abdellah Fourtassi and Emmanuel Dupoux. 2014. A Rudimentary Lexicon and Semantics Help Bootstrap Phoneme Acquisition. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 191–200, Ann Arbor, Michigan. Association for Computational Linguistics.

Stella Frank, Naomi H. Feldman, and Sharon Goldwater. 2014. Weak semantic context helps phonetic learning in a model of infant language acquisition. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Baltimore, Maryland. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *arXiv preprint*. ArXiv:2104.08821 [cs].

James Hillenbrand, Laura A. Getty, Kimberlee Wheeler, and Michael J. Clark. 1994. Acoustic characteristics of American English vowels. *The Journal of the Acoustical Society of America*, 95(5_Supplement):2875–2875.

Allard Jongman, Ratree Wayland, and Serena Wong. 2000. Acoustic characteristics of English fricatives. *The Journal of the Acoustical Society of America*, 108(3):1252–1263.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. *arXiv preprint*. Version Number: 5.

Patricia K. Kuhl. 2000. A new view of language acquisition. *Proceedings of the National Academy of Sciences*, 97(22):11850–11857.

Sang-Im Lee. 2011. Spectral analysis of Mandarin Chinese sibilant fricatives. In *Proceedings of the 17th international congress of phonetic sciences*, pages 1178–1181, Hong Kong.

Shanpeng Li and Wentao Gu. 2015. Acoustic analysis of Mandarin affricates. In *Interspeech 2015*, pages 1680–1684. ISCA.

Qisheng Liao, Meiting Lai, and Preslav Nakov. 2023. MarsEclipse at SemEval-2023 Task 3: Multi-lingual and Multi-label Framing Detection with Contrastive Learning. In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 83–87, Toronto, Canada. Association for Computational Linguistics.

Andrew Martin, Sharon Peperkamp, and Emmanuel Dupoux. 2013. Learning Phonemes With a Proto-Lexicon. *Cognitive Science*, 37(1):103–124.

Jessica Maye, Daniel J. Weiss, and Richard N. Aslin. 2008. Statistical phonetic learning in infants: facilitation and feature generalization. *Developmental Science*, 11(1):122–134.

Bob McMurray and Allard Jongman. 2011. What information is necessary for speech categorization? Harnessing variability in the speech signal by integrating cues computed relative to expectations. *Psychological Review*, 118(2):219–246.

Masaki Noguchi and Carla L. Hudson Kam. 2018. The Emergence of the Allophonic Perception of Unfamiliar Speech Sounds: The Effects of Contextual Distribution and Phonetic Naturalness. *Language Learning*, 68(1):147–176.

D Norris. 2003. Perceptual learning in speech. *Cognitive Psychology*, 47(2):204–238.

Judith E. Pegg and Janet F. Werker. 1997. Adult and infant perception of two English phones. *The Journal of the Acoustical Society of America*, 102(6):3742–3753.

Sharon Peperkamp, Rozenn Le Calvez, Jean-Pierre Nadal, and Emmanuel Dupoux. 2006. The acquisition of allophonic rules: Statistical learning with linguistic constraints. *Cognition*, 101(3):B31–B41.

Sharon Peperkamp, Michèle Pettinato, and Emmanuel Dupoux. 2003. Allophonic Variation and the Acquisition of Phoneme Categories. In *Proceedings of the 27th Annual Boston University Conference on Language Development*, volume 2, pages 650–661, Sommerville, MA. Cascadilla Press.

A. Seidl, A. Cristià, A. Bernard, and K. H. Onishi. 2009. Allophonic and Phonemic Contrasts in Infants' Learning of Sound Patterns. *Language Learning and Development*, 5(3):191–202.

Amanda Seidl and Alejandrina Cristia. 2012. Infants' Learning of Phonological Status. *Frontiers in Psychology*, 3.

Katrin Skoruppa, Anna Lambrechts, and Sharon Peperkamp. 2011. The role of phonetic distance in the acquisition of phonological alternations. In *Proceedings of the 39th Meeting of the North-East Linguistics Society (NELS)*.

D.H. Whalen, Catherine T. Best, and Julia R. Irwin. 1997. Lexical effects in the perception and production of American English /p/ allophones. *Journal of Phonetics*, 25(4):501–528.

A Supplementary materials

All scripts related to data manipulation, model architecture, loss function, and statistical analysis, along with additional resources, are accessible via the Open Science Framework: https://osf.io/krj2a/overview?view_only=d21b3d2a2ee64991a04f9ee456a93639.

Graded Expectations: Do Large Language Models Show Human-like Sensitivity to the Likelihood of Deceptive Speech Acts?

Xingyuan Zhao and Seana Coulson
University of California, San Diego
miozhao@ucsd.edu and scoulson@ucsd.edu

Abstract

Human discourse comprehension includes graded expectations about whether a speaker is likely to lie. If language models capture human-like discourse expectations, they should be sensitive not only to factual consistency but also to lie expectancy as a contextual probability from complex pragmatic cues. We test this idea using discourse scenarios with varying incentives to deceive. Human lie probability is estimated from free continuations, and model lie expectancy is derived from the probability mass assigned to human-produced lie versus truth continuations. Across Qwen3 models, likelihood-derived lie mass aligns strongly with human lie expectancy. The best performance comes from the base checkpoints. By contrast, post-trained and mode-specialized variants show weaker alignment. Qualitative analysis suggests a structured error pattern: models tend to overpredict lies when a response directly conflicts with known facts, but underpredict them when lie expectancy depends more on contextual pressures such as politeness, self-protection, or strategic gain. These results suggest that graded lie expectancy is recoverable from model continuation probabilities and can be learned, at least in part, through the ordinary next-token prediction objective.

1 Introduction

Human discourse comprehension is not limited to recovering the literal meaning of an utterance. In context, comprehenders also form expectations about why a speaker is saying something, whether the speaker is likely to be truthful, and how social or situational pressures shape the likely continuation of the interaction (Grice, 1975; Van Berkum, 2009; Hagoort et al., 2004). Lie expectancy is therefore not a peripheral phenomenon. It is part of ordinary discourse understanding: observers often anticipate that a speaker will lie in a given situation, even when they themselves know the truth, because they can infer the speaker's motives from the surrounding context.

This makes lie probability a meaningful target for language models. If a model captures human-like discourse expectations, it should be sensitive not only to factual consistency, but also to when deception is more or less likely in context. Recent work has shown that language models exhibit certain pragmatic abilities, including interpreting non-literal language and reasoning about speaker intent (Hu et al., 2023), yet their performance on tasks requiring inference about others' mental states remains mixed (Strachan et al., 2024; Kosinski, 2024; Sap et al., 2022; Ullman, 2023). Lies are shaped by structured pressures: avoiding punishment, protecting self-image, sparing another person's feelings, securing material gain, or manipulating what the listener will believe or do. These pressures depend on more than isolated lexical cues. They arise from the interaction of discourse facts, speaker goals, social relationships, institutional settings, and culturally familiar patterns of behavior. A model that is insensitive to this structure may detect contradiction, yet still fail to approximate human expectations about when a lie is likely.

At the same time, lie expectancy is not the same as binary lie detection. Determining that a particular response is false relative to a scenario is a simpler problem than estimating how likely a speaker is to lie in that scenario. The former can often be supported by recovering the relevant facts and detecting inconsistency. The latter is more demanding: it requires a graded expectation over possible continuations and depends on complex pragmatic factors. This distinction is central to the present study. Our aim is not merely to ask whether language models can identify a lie, but whether their continuation preferences recover the *graded human expectation* that a speaker will lie.

This framing motivates a likelihood-based approach. If lie expectancy is part of discourse pre-

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 487–495
July 3-4, 2026 ©2026 Association for Computational Linguistics

diction, then it should be reflected in the probability distribution a model assigns to possible continuations. Rather than relying on explicit metalinguistic prompting—which has been shown to diverge from direct probability measurements, particularly for complex linguistic tasks (Hu and Levy, 2023)–we ask whether a model assigns greater probability mass to human-produced lie continuations in contexts where humans themselves are more likely to expect a lie. Under this view, deception expectancy is not a special-purpose judgment layered on top of language processing; it is part of the predictive structure of discourse itself.

To test this idea, we use discourse scenarios in which speakers have varying motivations to lie, and we estimate human lie probability from free completions collected for a subset of items. We then derive model lie expectancy from the probability mass assigned to lie-labeled versus truth-labeled human continuations. Because this measure is defined over continuation probabilities, it directly targets the model's latent predictive structure rather than its explicit verbal judgments.

We evaluate Qwen3 models (Qwen Team, 2025) across multiple scales and training conditions, including base checkpoints and post-trained checkpoints. Before the main alignment analysis, we also assess preliminary competence on lie–truth classification and discourse comprehension tasks, allowing us to distinguish competence of basic contextual understanding from competence to recover graded human lie expectancy.

The central question of the paper is therefore straightforward: *do language models capture human graded expectations about when a speaker will lie?* More specifically, we ask whether lie expectancy is recoverable from model continuation distributions, whether this alignment varies as a function of scale and post-training, and whether stronger general competence necessarily implies stronger alignment with human lie expectancy. Across the Qwen models evaluated here, we find that likelihood-derived lie mass aligns strongly with human lie expectancy, with the best performance coming from larger base checkpoints. This suggests that base models can align more closely with human behavior than their post-trained counterparts (Kim and Davis, 2025), and graded lie expectancy is, at least in part, a kind of discourse-level predictive structure that language models can acquire through next-token prediction over distributional regularities in context.

2 Dataset and Human Measures

2.1 Materials

The materials used here were adapted from an ongoing study investigating the online processing of lies and truths in socially biased discourse contexts. Each item consists of a short narrative scenario that establishes a social situation in which a speaker has a clear motivation to lie (e.g., to avoid punishment, to spare someone's feelings, or to appear favorably to others). The scenario is followed by a target sentence containing a critical word whose substitution determines whether the utterance is a lie or a truth relative to the scenario. Critical words occur in antonym pairs (e.g., *boring/interesting*), and the scenarios are likewise paired such that a critical word that produces a lie in one scenario produces a truth in the other, so that differences between conditions are not attributable to lexical properties of the critical words.

Example A:

> *While hanging out at the mall, Jake saw a beautiful leather jacket and decided to take it. He put it on and walked quickly out of the store. But, as Jake passed through the door, an alarm went off. A security guard stopped him and asked him where he got the jacket.*
>
> `Lie: Jake told the guard that he just bought the new jacket.`
>
> `Truth: Jake told the guard that he just took the new jacket.`

Example B:

> *Pete had just bought a brand new cellphone and was excited to show it to his friends. He knew they would be impressed if they thought he had taken it from the store without paying. When he showed it to them, they asked Pete where he got the cellphone.*
>
> `Lie: Pete told his friends that he just took the new cellphone.`
>
> `Truth: Pete told his friends that he just bought the new cellphone.`

The full stimulus set comprises 76 scenarios, each ending with one of two potential target sentences (a lie or a truth), and followed by one of the two possible probes (a plausible or an implausible

probe continuation). For example, following the lie in the sample stimulus in EXAMPLE A, the plausible continuation probe was "The security guard asked Jake to show him the receipt for the jacket," and the implausible continuation was "The security guard invited Jake to a slumber party." All 76 items are used in the preliminary competence evaluations (Experiment 1). We then collected and annotated free completions based on the scenario from 44 participants for a subset of 40 items, and we use this as the human lie expectancy in modeling the alignment between LLM lie mass and human lie expectancy(Experiment 2).

3 Language Model Selection

We evaluate models from the Qwen3 family (Qwen Team, 2025) in order to examine how alignment with human lie expectancy varies as a function of model scale and post-training regime.

We compare models across four scales (0.6B, 1.7B, 4B, and 8B) and across three training conditions: base, post-trained, and later checkpoint variants.

Base models. (*Qwen3-0.6B-Base*, *Qwen3-1.7B-Base*, *Qwen3-4B-Base*, *Qwen3-8B-Base*). Base models derive their continuation preferences from distributional semantics in the pre-training corpus, without the additional reshaping introduced by later post-training. Their next-token distributions should therefore provide the cleanest estimate of the model's underlying continuation preferences.

Post-trained models. (*Qwen3-0.6B*, *Qwen3-1.7B*, *Qwen3-4B*, *Qwen3-8B*). These models have undergone multi-stage post-training, including reasoning-oriented optimization, Thinking Mode Fusion, and general reinforcement learning. Thinking Mode Fusion integrates thinking and non-thinking behaviors within a single checkpoint. Such post-training may improve downstream capability while also reshaping token-level continuation probabilities. Comparing them against the base models therefore tests whether post-training preserves or distorts the continuation structure relevant to human lie expectancy.

4B checkpoint variants. (*Qwen3-4B-Thinking-2507*, *Qwen3-4B-Instruct-2507*). These checkpoints allow a more fine-grained comparison within the 4B line. Both are causal language models released after pretraining and post-training. Unlike the original post-trained Qwen3 models, `Qwen3-4B-Thinking-2507` supports only thinking mode, automatically inserts `<think>` in the default chat template, and is described as having increased thinking length for highly complex reasoning tasks. `Qwen3-4B-Instruct-2507`, in contrast, supports only non-thinking mode and does not generate `<think></think>` blocks. A dual-mode fused model may encode a continuation distribution that reflects a compromise between deliberative reasoning-style behavior and direct response behavior. A single-mode checkpoint, by contrast, should express a more specialized continuation distribution.

This sampling scheme distinguishes between two possibilities. Human-like lie expectancy may be best preserved in pretraining-induced continuation structure, in which case base models should align most closely with human behavior. Alternatively, post-training may sharpen the pragmatic and inferential structure relevant to deception, in which case post-trained or thinking-oriented variants should perform better. A further possibility is that post-training improves general capability while distorting the token-level continuation probabilities on which our measure depends.

4 Preliminary Competence Evaluation: Experiment 1

Before testing item-level alignment with human lie expectancy, we asked whether the evaluated models possess basic competence on two discourse comprehension tasks. The first is *lie/truth discrimination*: given a scenario and a candidate utterance, the model must determine whether the utterance constitutes a lie or a truth in that context. The second task is a *scenario coherence judgment*: given a preceding scenario and a probe sentence, the model must determine whether the probe is a plausible continuation of the story. While not the primary theoretical target of the paper, these tasks serve as diagnostic checks on the models' capacity to represent the facts presented in the scenarios. Accordingly, performance on these preliminary tasks will allow us to determine whether subsequent failures in the main alignment analyses are attributable to lack of contextual understanding.

4.1 Lie–truth discrimination

We first evaluated whether each model could distinguish lies from truths relative to the discourse scenario. For each item, the model was shown the

scenario together with a candidate target utterance and asked to classify the target as either a *lie* or a *truth*. This evaluation was conducted on the full 76-item stimulus set. Classification accuracy was initially computed across all items, and then recorded separately for lie and truth target sentences. We reasoned that a model that cannot reliably distinguish a lie from the truth would be unlikely to succeed on the graded lie-expectancy analyses in Experiment 2. The purpose of this discrimination task was thus diagnostic and served as a preliminary check on contextual understanding.

Qwen-family results are shown in Table 1 and differed markedly across the models. Several smaller or no-thinking configurations showed highly asymmetric behavior, often with inflated lie accuracy coupled with very poor truth accuracy. For example, in no-thinking mode, Qwen3-0.6B achieved chance-level overall accuracy (.500) with perfect lie accuracy and zero truth accuracy, indicating a strong bias to respond "lie." A similar pattern appeared in several base and no-thinking configurations. By contrast, thinking-enabled and larger variants performed far better than their base and no-thinking counterparts. The strongest overall performance came from Qwen3-8B in thinking mode (.882), followed by Qwen3-4B-Thinking-2507 (.848) and Qwen3-4B in thinking mode (.829). These results indicate that while many Qwen configurations possess basic contextual competence, this competence depends strongly on model scale and post-training regime.

4.2 Scenario coherence judgment

We next evaluated whether the models could distinguish contextually coherent from incoherent probe continuations. For each item, the model was given the scenario and a probe sentence and was asked to judge whether the probe was coherent with the discourse context. Because each scenario was paired with plausible and implausible probes in both lie and truth conditions, this task supplements the binary lie/truth task and provides a broader diagnostic of discourse-level comprehension. Accuracy was computed both overall and separately for plausible versus implausible probes within the lie and truth conditions, respectively.

Figure 1 shows that human comprehension performance remained substantially above all evaluated Qwen models. Human accuracy was near ceiling overall (All = .912), with high performance in both lie (.920) and truth (.904) conditions. Among Qwen models, performance varied by scale and mode: base variants were strongest in the 0.6B, 4B, and 8B families, whereas the 1.7B family showed a weaker and less consistent pattern. The best-performing models were Qwen3-4B-Base (.789), Qwen3-4B-Thinking-ckpt (.783), and Qwen3-8B-Base (.776), but all remained below the human benchmark. This confirms that even the strongest Qwen variants exhibit a substantial gap from human comprehension on the preliminary discourse task.

Taken together, the results of Experiment 1 suggest that a subset of the models possess nontrivial discourse competence on the materials used here. At the same time, the substantial variability across Qwen configurations indicates that model family, scale, and post-training regime materially affect even these prerequisite abilities. In Experiment 2 we go on to ask whether the representations that develop in these models align with *graded* human expectations that a given speaker will lie.

5 Does Model Lie Mass Align with Human Lie Expectancy? Experiment 2

Experiment 2 asks whether models' continuation probabilities align with graded human lie expectancy at the item level.

5.1 Human lie probability

For a subset of 40 items, we collected free completions from 44 participants. Each participant read the scenario and the target sentence prefix (up to but not including the critical word) and provided a free-form completion. The first author then coded each completion as a lie response, a truth response, or ambiguous/not applicable.

For each item i, human lie probability is defined as the proportion of lie responses among all unambiguous lie/truth responses:

$$\hat{p}_i^{\text{human}}(\text{lie}) = \frac{n_i^L}{n_i^L + n_i^T}, \tag{1}$$

where n_i^L and n_i^T denote the number of lie and truth responses for item i, excluding ambiguous/not applicable responses from the denominator. This quantity serves as the human target in the item-level alignment analyses reported in this experiment.

In addition, for each item we retain the full set of observed human completions $\mathcal{C}_i$, together with their lie/truth labels and multiplicity counts $m(c)$

Model	Condition	Accuracy	Acc_{Lie}	$\text{Acc}_{\text{Truth}}$
Qwen3-0.6B	instruct, no-thinking	0.500	1.000	0.000
Qwen3-0.6B	instruct, thinking	0.645	0.737	0.553
Qwen3-0.6B-Base	base	0.487	0.974	0.000
Qwen3-1.7B	instruct, no-thinking	0.493	0.947	0.039
Qwen3-1.7B	instruct, thinking	0.755	0.868	0.640
Qwen3-1.7B-Base	base	0.500	0.789	0.211
Qwen3-4B	instruct, no-thinking	0.539	0.789	0.289
Qwen3-4B	instruct, thinking	0.829	0.934	0.724
Qwen3-4B-Base	base	0.592	0.947	0.237
Qwen3-4B-Thinking-2507	thinking checkpoint	0.848	0.974	0.720
Qwen3-4B-Instruct-2507	instruct checkpoint	0.664	0.842	0.487
Qwen3-8B	instruct, no-thinking	0.612	0.750	0.474
Qwen3-8B	instruct, thinking	**0.882**	0.974	0.789
Qwen3-8B-Base	base	0.724	0.829	0.618

Table 1: Preliminary lie–truth discrimination results for Qwen-family models. "Condition" distinguishes decoding or model configuration conditions. Thinking-mode Qwen models generally outperform matched no-thinking and base variants, with the strongest overall performance from Qwen3-8B in thinking mode.

for each unique completion $c \in \mathcal{C}_i$. These completions serve as the candidate set for the model likelihood-based lie mass measure: rather than generating continuations freely, the model scores the probability of each human-produced completion, weighted by the number of participants who produced it.

5.2 Model Likelihood-Based Lie Mass

Rather than eliciting an explicit judgment from the model, we derive *model* lie expectancy from the probability mass assigned to human-produced lie versus truth continuations. For each item i, we construct a context as in (2).

$$x_i = \text{scenario}_i + \text{prefix}_i. \tag{2}$$

Let C_i be the set of human completions for item i. For each completion $c \in C_i$, we compute a length-normalized conditional log-likelihood:

$$s(c \mid x_i) = \frac{1}{|c|} \sum_t \log P(w_t \mid x_i, w_{<t}), \tag{3}$$

where $|c|$ is the number of tokens in completion c, and w_t is the t-th token of that completion. To convert these scores into relative weights over the observed completion set, we apply a count-weighted softmax:

$$\tilde{q}(c \mid x_i) = \frac{m(c) \exp(s(c \mid x_i))}{\sum_{c' \in C_i} m(c') \exp(s(c' \mid x_i))}, \tag{4}$$

where $m(c)$ is the number of human participants who produced completion c. We then define model lie probability as the total normalized mass assigned to lie-labeled completions:

$$\hat{p}_i^{\text{model}}(\text{lie}) = \sum_{c \in C_i^L} \tilde{q}(c \mid x_i), \tag{5}$$

where $C_i^L \subseteq C_i$ denotes the subset of lie-labeled completions. This measure is intended to capture implicit deception expectancy as encoded in the model's continuation preferences, rather than in its explicit verbal judgments.

We evaluate alignment between model-derived lie probability and human lie probability using four item-level metrics: Pearson correlation, Spearman correlation, mean absolute error (MAE), and root mean squared error (RMSE). Pearson correlation measures linear correspondence to human lie expectancy, whereas Spearman correlation measures rank-order agreement. MAE and RMSE quantify absolute deviation from the human values, with lower values indicating closer fit.

5.3 Item-Level Alignment Results

Table 2 reports alignment results for all Qwen-family models. Across models, likelihood-derived lie probability shows substantial positive correspondence with human lie expectancy, indicating that model continuation distributions recover graded structure in human deception expectations.

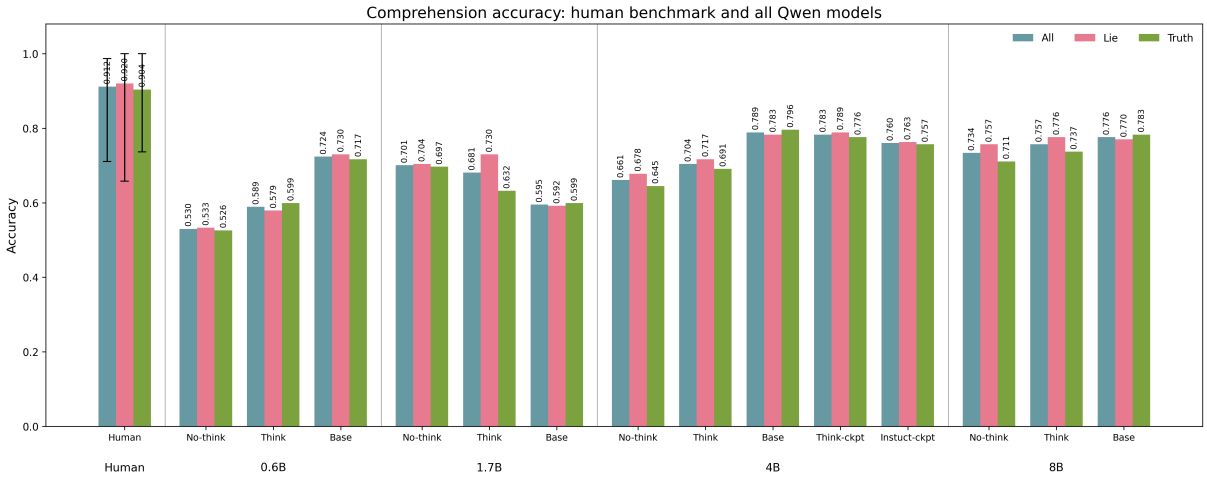


Figure 1: Human benchmark and Qwen-model performance on the preliminary comprehension task. Bars show overall accuracy (All), lie-condition accuracy (Lie), and truth-condition accuracy (Truth). Human values are means across participants, and error bars indicate the observed min–max range across participants with non-missing comprehension scores. Qwen models are grouped by parameter scale and mode.

Model	Pearson r	Spearman ρ	MAE	RMSE
4B-Base	**0.838**	**0.892**	0.166	0.233
8B-Base	0.806	0.852	**0.146**	**0.227**
0.6B-Base	0.788	0.829	0.173	0.260
8B	0.716	0.808	0.231	0.312
1.7B-Base	0.745	0.801	0.182	0.263
4B-Thinking-ckpt	0.728	0.797	0.213	0.299
0.6B	0.714	0.781	0.191	0.270
4B-Instruct-ckpt	0.690	0.780	0.231	0.325
4B	0.688	0.780	0.237	0.333
1.7B	0.611	0.720	0.250	0.360

Table 2: Likelihood-based alignment results for Qwen-family models on all items. Pearson and Spearman quantify item-level alignment with human lie probability; lower MAE and RMSE indicate closer absolute fit.

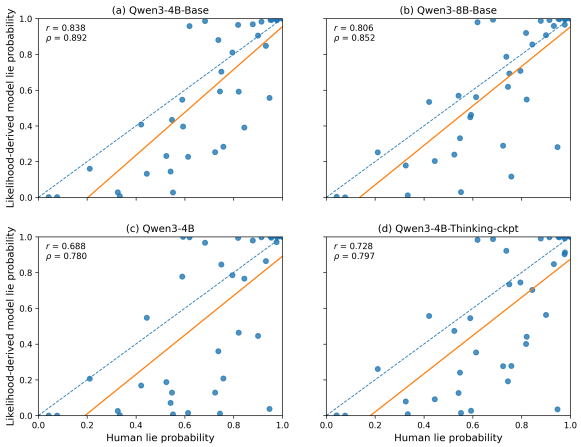


Figure 2: Item-level relationship between human lie probability and likelihood-derived model lie probability for representative Qwen models. The top row shows two base checkpoints (`Qwen3-4B-Base` and `Qwen3-8B-Base`), and the bottom row shows two non-base 4B variants (`Qwen3-4B` and `Qwen3-4B-Thinking-2507-checkpoint`). The dashed diagonal indicates perfect agreement ($y = x$), and the solid line shows the fitted linear trend.

The strongest performance is observed in the base checkpoints. In particular, `Qwen3-4B-Base` yields the strongest item-level alignment by correlation, achieving the highest Pearson r and Spearman ρ, whereas `Qwen3-8B-Base` yields the lowest absolute error, achieving the lowest MAE and RMSE.

More generally, base models show stronger alignment and lower absolute error than their post-trained counterparts at comparable scales. This pattern suggests that human-like lie expectancy is more faithfully preserved in pretraining-induced continuation structure than in distributions reshaped by later post-training.

Figure 2 illustrates the item-level relationship between model lie probability and human lie probability for representative models. Base checkpoints exhibit tighter alignment and less dispersion, whereas non-base variants show larger absolute deviations from the human values. Across panels, the fitted lines generally fall below the identity line, indicating systematic underestimation: the models track which items are more lie-favoring, but assign less extreme probabilities than humans do.

5.4 Qualitative analysis of high-disagreement items

We then took a closer look at the shared outliers among the best aligned Qwen models. The two

strongest shared overestimation items involve locally explicit contradictions between the discourse context and the speaker's response, and in both cases the models assign near-ceiling lie probability. In `overestimation items 1`, Brenda forgets to lock the door before a burglary. Her parents asked her if she had secured the door before leaving home. In `overestimation items 2`, Nancy has an eating disorder. She has eaten only an apple. Her doctor asks her what she ate that day. Why do the Qwen models expect lies in these scenarios when the humans do not? One possible explanation is that the models overweight the inferential structure in the scenario without fully capturing contextual pressures toward honesty(Sap et al., 2022; Ullman, 2023). The Brenda scenario, for example, includes explicit mention of her failure to lock the door followed by her parents query about this specific possibility. Moral norms surrounding interactions with one's parents may influence humans to entertain the possibility that Brenda might confess. By contrast, the models treat the conflict between the unlocked door implied by the first sentence and the locked door presupposed in the second as strongly predictive of a deceptive speech act. Likewise, whereas Nancy's eating disorder suggests she is likely to lie about her diet, the clinical interaction with her doctor may carry a competing norm of truthful disclosure. On this view, the models overestimate lie probability when local contradiction is strong but the social or institutional context leave room for honesty.

The shared underestimation items show the opposite pattern as the models underweight the social motivations to lie in the absence of conflicting factual information. In `underestimation items 1`, Rachel buys a painting at a low price. After it is damaged, the insurance agent asks how much she paid for the painting. In `underestimation items 2`, Brittany privately dislikes her friend's wedding dress, but knows it is too late to return the gown. Given she doesn't want to upset her friend, her friend asked Brittany what she thought of the dress. In both cases, the models assign substantially lower lie probability than human participants do. Unlike the overestimation items, the deceptive responses here are signaled less by blunt contradiction with an explicitly stated fact than by a socially or strategically motivated choice of utterance. In `underestimation items 1`, the relevant pressure comes from financial self-interest in an institutional setting; in `underestimation items 2`, it comes from face-saving politeness and the desire to avoid hurting another person's feelings. One possible interpretation is that the models are less sensitive to these higher-order pragmatic pressures than to direct factual conflict. On this view, the models underestimate lie probability when deception depends on social or strategic motivation rather than on an immediately salient contradiction with the discourse context.

Taken together, these cases suggest a common error pattern. The models are highly sensitive to explicit contradiction with known facts, but they underweight contextual factors that shape whether a speaker is expected to tell the truth or lie in a particular social setting. These interpretations remain tentative, but they suggest that model–human disagreement in the present task may arise in part from differences in how local factual conflict and higher-order pragmatic pressures are weighted.

6 Is Lie-Expectancy Alignment Reducible to General Competence?

A natural question is whether downstream alignment with human lie expectancy can be reduced to more general model competence. To examine this possibility, we conducted an exploratory cross-model comparison relating each Qwen model's Experiment 1 performance to its Experiment 2 alignment with human lie expectancy(Figure 3).

The two preliminary measures showed different patterns. Lie-classification accuracy was weakly negatively associated with downstream alignment, indicating that models that were better at binary lie–truth discrimination were not necessarily better at recovering graded human lie expectancy. By contrast, comprehension accuracy showed a moderate positive association with downstream alignment, suggesting that broader discourse-level understanding may be more relevant to human-like lie expectancy than categorical lie labeling alone.

These comparisons should be interpreted cautiously given the small number of models. Still, the pattern is informative. A plausible explanation is that binary lie–truth classification and lie-expectancy alignment rely on different levels of representation. Binary classification can be supported by recovering the relevant discourse facts in the scenario and determining whether a response is consistent with them. Lie-expectancy alignment is more demanding: it requires not only detecting factual inconsistency, but also modeling how likely

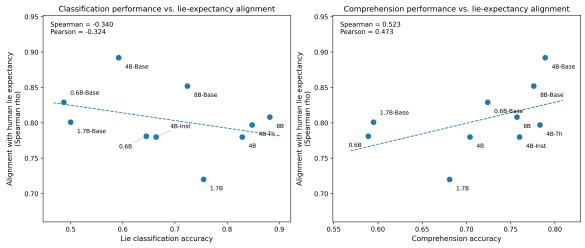


Figure 3: Exploratory cross-model relationship between preliminary task performance and downstream alignment with human lie expectancy in Qwen models. Left: lie-classification accuracy versus alignment with human lie expectancy. Right: comprehension accuracy versus alignment with human lie expectancy. Alignment is indexed by Spearman correlation between model-based lie mass and human lie probability. Dashed lines show least-squares fits for visualization only.

a speaker is to lie in that particular context. That, in turn, depends on pragmatic and discourse-level factors such as speaker goals, situational incentives, social pressures, and the relative plausibility of lie-favoring versus truth-favoring continuations. On this view, it is not surprising that downstream alignment is more closely related to comprehension accuracy than to binary lie-classification accuracy.

7 Conclusion

This paper asked whether language models capture graded human expectations about when a speaker is likely to lie in context. Across Qwen3 models, lie expectancy was recoverable from continuation probabilities over human-produced completions. The strongest alignment came from the base checkpoints, showing that stronger performance on post-trained or explicit task-style settings does not automatically yield more human-like lie expectancy.

The relevant signal depends on discourse-level context: model probabilities are conditioned on the scenario and utterance prefix rather than on isolated lexical material. The structured error pattern supports the same conclusion. Model errors were not random: the models were more sensitive to fact–response inconsistency than to contextual pressures such as politeness, accountability, or strategic self-interest. These findings are broadly compatible with psycholinguistic views in which comprehenders build structured discourse or situation-level representations that guide expectations about upcoming language (Zwaan and Radvansky, 1998; Kuperberg and Jaeger, 2016) .

The study also has clear limitations. The main results come from one model family, a relatively small item set, and a likelihood-based measure defined over observed human completions rather than unrestricted generation. Future work should test whether the same pattern holds across other architectures and should relate model errors more systematically to specific discourse variables, such as face-saving pressure, institutional incentives, and asymmetries in shared knowledge.

In sum, we find that likelihood-derived lie mass derived from the Qwen models aligns strongly with human lie expectancy, with the best performance coming from larger base checkpoints. This suggests that human-like lie expectancy is learnable, at least to some extent, from next-token prediction, not as an explicit theory of deception, but as a statistical regularity distributed across discourse, pragmatic, and situational structure.

References

H. Paul Grice. 1975. Logic and conversation. In *Speech Acts*, pages 41–58. Brill.

Peter Hagoort, Lea Hald, Marcel Bastiaansen, and Karl Magnus Petersson. 2004. Integration of word meaning and world knowledge in language comprehension. *Science*, 304(5669):438–441.

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.

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*, pages 5040–5060, Singapore. Association for Computational Linguistics.

Sanghee J. Kim and Forrest Davis. 2025. Discourse sensitivity in attraction effects: The interplay between language model size and training data. In *Proceedings of the Society for Computation in Linguistics*.

Michal Kosinski. 2024. Evaluating large language models in theory of mind tasks. *Proceedings of the National Academy of Sciences*, 121(45).

Gina R Kuperberg and T Florian Jaeger. 2016. What do we mean by prediction in language comprehension? *Language, cognition and neuroscience*, 31(1):32–59.

Qwen Team. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Maarten Sap, Ronan LeBras, Daniel Fried, and Yejin Choi. 2022. Neural theory-of-mind? On the limits of social intelligence in large LMs. *arXiv preprint arXiv:2210.13312*.

James W. A. Strachan, Dalila Albergo, Giulia Borghini, Oriel Pansardi, Eugenio Scalber, Alessandro Ruber, Guido Manzi, Matilde De Luca, Gabriella Recchia, and Cristina Becchio. 2024. Testing theory of mind in large language models and humans. *Nature Human Behaviour*, 8(7):1285–1295.

Tomer Ullman. 2023. Large language models fail on trivial alterations to theory-of-mind tasks. *arXiv preprint arXiv:2302.08399*.

Jos JA Van Berkum. 2009. The neuropragmatics of'simple'utterance comprehension: An erp review. In *Semantics and pragmatics: From experiment to theory*, pages 276–316. Palgrave Macmillan.

Rolf A. Zwaan and Gabriel A. Radvansky. 1998. Situation models in language comprehension and memory. *Psychological Bulletin*, 123(2):162–185.

Lexical exceptionality in paradigm-specific learning: modeling stem-final obstruent alternations in Korean verbs and adjectives

Stella Eunsoo Hong
University of Utah
u1597092@utah.edu

Abstract

Korean stem-final conjugations illustrate the interaction between lexical exceptionality and heterogeneous phonological processes. When /p/-, /t/-, and /s/-final stems occur before vowel-initial suffixes, the irregular classes in these paradigms undergo intervocalic lenition, each exhibiting a distinct alternation pattern. Learners must therefore not only identify which roots trigger lenition, but also determine the corresponding repair strategy. This study investigates how lexically-specific phonological patterns are acquired when multiple repair strategies are available. We employ a lexically scaled MaxEnt model (Linzen et al., 2013; Hughto et al., 2019) to learn these paradigm-specific alternations and run simulations under two learning scenarios: (1) when repair strategies occur at equal frequencies and (2) when one strategy significantly outnumbers the others. Results show that the model favors a *least-cost* solution by treating statistically dominant morpheme classes as the general pattern. We conclude by discussing the model's sensitivity to lexical statistics, predictions for empirical testing, and implications for language acquisition.

1 Introduction

Lexical exceptionality requires learners to reconcile competing generalizations within a single grammatical system. When exceptional morphemes resist the general pattern, they challenge the constraint rankings that otherwise account for the majority of forms. Within Optimality Theory (OT), two core approaches to modeling such conflicts are cophonology (Inkelas et al., 1997; Anttila, 2002; Inkelas and Zoll, 2003) and lexical indexation (Pater, 2000; Coetzee, 2009). Cophonology assumes separate rankings for different morphological classes, thereby permitting multiple subgrammars to resolve the conflict. Lexical indexation, by contrast, indexes constraints to specific lexical classes, allowing a single ranking to accommodate the conflict through lexically restricted application of the constraint.

Previous literature has largely focused on exceptionality involving a single phonological process. Less attention has been paid to cases in which multiple repair processes coexist, which can potentially introduce substantial ranking complexity under both approaches introduced above. Nevertheless, many languages employ more than one strategy to repair ill-formed phonological sequences (Pater, 1999). When such competing repair strategies intersect with lexical exceptionality, learners must determine not only how exceptional and non-exceptional items are represented, but also how competing repairs are ranked against one another. This raises the question of how learners reconcile inconsistency across two dimensions: lexical specificity and competing repair strategies.

Korean stem-final conjugations exemplify such empirical patterns and serve as a useful test case for this question. When irregular classes of verbs and adjectives ending in /p/, /t/, and /s/ combine with vowel-initial suffixes, the stem-final obstruents surface as [w], [ɾ], and ∅, respectively, through intervocalic lenition. All three lenition strategies are motivated by avoidance of the dispreferred *V-OBS-V sequence, but each applies only to a specific stem-final paradigm.

To address the learning of *paradigm-specific lexical exceptionality*, we adopt a lexically scaled MaxEnt framework (Linzen et al., 2013; Hughto et al., 2019). As a probabilistic implementation of the indexation approach, the model encodes both lexical exceptionality and competing repair strategies within a single set of weights. Lexical specificity is implemented through additive morpheme-specific weights associated with particular paradigm classes, thereby capturing the role of indexed constraints while avoiding the need to posit separate subgrammars.

Moreover, the distribution of irregular forms in

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 496–512
July 3-4, 2026 ©2026 Association for Computational Linguistics

Korean varies across stem classes, allowing us to examine how statistical properties influence the choice of repair strategy. The majority of /p/-final stems exhibit irregular alternations, whereas /t/- and /s/-final stems show roughly equal proportions of regular and irregular forms. To investigate the effect of this distributional asymmetry, we ran a toy simulation under two learning scenarios: (1) when only /t/- and /s/-final paradigms with roughly equal proportions of regular and irregular forms are considered, and (2) when the /p/-final paradigm, which exhibits a strong majority of irregular forms, is added. For both experimental conditions, the number of suffixes was incrementally increased to explore how the degree of affix exposure influences success in learning stem-specific exceptionality.

Results show improved generalization as the number of suffix types increases.[1] When given only a few suffix types, additive morpheme-specific weights were incorrectly assigned to suffixes. As the suffix inventory grew, these weights gradually shifted toward stems, correctly identifying them as the locus of lexical exceptionality. This process first involved assigning scaling weights selectively to irregular classes, thereby distinguishing them from regular classes, and was later refined as the model began to capture paradigm-specific distinctions among irregular forms. In the experiment including the /p/-final paradigm, the learner treated the /p/-irregular alternation as the default repair when lenition was triggered, as the constraint violated by this class showed the greatest resistance to receiving general weight. This configuration came at the cost of reduced prediction accuracy for the other two alternating classes.

The rest of the paper is organized as follows. Section 2 introduces the empirical patterns. Section 3 reviews the relevant literature and motivates the use of lexically scaled MaxEnt to model Korean stem-final conjugation. Section 4 presents the implementation details of the simulations, and Section 5 summarizes the results. Finally, Section 6 discusses the implications of these findings and outlines how learning may proceed when real-world data are fully incorporated.

[1]Our experimental code and data are available at https://github.com/stxllastar/scaled_conjugation.

2 Korean stem-final verb / adjective conjugations

Korean verbs and adjectives are traditionally divided into regular and irregular classes. Regularity is defined in terms of form invariance: regular forms remain unchanged within a paradigm, whereas irregular forms exhibit alternation before certain suffixes (Lee, 1996; Rhee, 1996). For clarity, we refer to irregular and regular classes as alternating and non-alternating classes, respectively[2].

Alternating conjugation patterns can be further classified into three types depending on the locus of alternation: (1) the stem, (2) the suffix (i.e., the inflectional ending), or (3) both. Within the first type, /p/-, /t/-, and /s/-final stems constitute a particularly interesting case, as their alternations involve intervocalic lenition conditioned by the same phonological environment.

(1) Intervocalic lenition in alternating classes
   a. p-stems: /p/ → [w] / V_V
   b. t-stems: /t/ → [ɾ] / V_V
   c. s-stems: /s/ → ∅ / V_V

As (1) illustrates, when vowel-initial suffixes follow alternating-class stems, the stem-final obstruents /p/, /t/, and /s/ undergo lenition, surfacing as the sonorants [w] and [ɾ] or deleting (Lee, 2013; Albright and Kang, 2009). Because Korean syllable structure disallows complex onsets and codas, vowel-initial suffixation places stem-final obstruents in an intervocalic environment. This lenition process is lexically restricted, as stems belonging to the non-alternating class remain invariant in the same environment—a contrast illustrated in Table 1.

Class	Stem	Conjugation
p-alternating	/ka.ka**p**/ 'close'	[ka.ka.**w**-ʌ]
p-non-alternating	/i**p**/ 'wear'	[i.**p**-ʌ]
t-alternating	/kjɛ.da**t**/ 'realize'	[kjɛ.da.**ɾ**-a.jo]
t-non-alternating	/mu**t**/ 'bury'	[mu.**t**-ʌ]
s-alternating	/na**s**/ 'better'	[na.-a.do]
s-non-alternating	/u**s**/ 'laugh'	[u.**s**-ʌ]

Table 1: Alternating vs Non-alternating classes in V_V

Meanwhile, when stems are followed by consonant-initial suffixes, the environment triggering alternation is absent. Consequently, the contrast between classes observed in Table 1 disappears: the

[2]The traditional distinction between regular and irregular classes does not directly correspond to phonological regularity, as the forms labeled *irregular* undergo systematic alternations and in fact outnumber the regular forms.

underlying /p/, /t/, and /s/[3] surface faithfully in both classes. As shown in Table 2, both alternating and non-alternating stems preserve their underlying codas before the declarative suffix *-ta*.

Class	Stem	Conjugation
p-alternating	/ka.ka**p**/ 'close'	[ka.ka**p**.-ta]
p-non-alternating	/i**p**/ 'wear'	[i**p**.-ta]
t-alternating	/kjɛ.da**t**/ 'realize'	[kjɛ.da**t**.-ta]
t-non-alternating	/mu**t**/ 'bury'	[mu**t**.-ta]
s-alternating	/na**s**/ 'better'	[na**s**.-ta]
s-non-alternating	/u**s**/ 'laugh'	[u**s**.-ta]

Table 2: Alternating vs Non-alternating classes in V_C

To summarize, the phenomenon exhibits three key properties. First, it shows lexical idiosyncrasy: only alternating (irregular) stems undergo lenition. Second, it involves paradigm-specific repair, whereby each obstruent-final alternating class adopts a distinct lenition strategy to avoid the *V-OBS-V configuration. Third, the process reflects stem-based exceptionality, with suffixes merely providing the phonological environment that triggers the alternation (intervocalic with -V suffixes and non-intervocalic with -CV suffixes). Hereafter, we refer to suffixes as *environmental triggers*.

3 Previous approaches towards learning lexical specificity

3.1 The lexical indexation approach

Among the two widely discussed approaches to lexical exceptionality—cophonology and lexical indexation—Korean conjugation has been analyzed within the framework of lexical indexation. Lee (2013) adopts the Recursive Constraint Demotion (RCD) algorithm (Tesar and Smolensky, 1998; Tesar and Prince, 2003) in conjunction with constraint cloning (Pater, 2009) to address the ranking conflict between alternating and non-alternating classes within each paradigm.

RCD incrementally constructs a constraint hierarchy by installing the best-performing constraint—one that prefers only winning candidates—immediately below the established ranking. However, in cases of lexical exceptionality, no single constraint ordering satisfies all winner–loser comparisons: ranking *V-OBS-V below the faithfulness constraint correctly favors the non-alternating-class winner but incorrectly prefers the losing candidate for the alternating class, whereas ranking *V-OBS-V above the faithfulness constraint produces the opposite result.

The conflicting requirements for alternating stems (*V-OBS-V ≫ FAITH) and non-alternating stems (FAITH ≫ *V-OBS-V) are resolved by cloning *V-OBS-V and indexing its applicability separately to the two classes. This yields the ranking scheme in (1), where both orderings are satisfied through a partial ordering of class-specific constraints.

$$*\text{V-OBS-V}_{\text{Alter}} \gg \text{FAITH} \gg *\text{V-OBS-V}_{\text{Non-Alter}} \quad (1)$$

What remains unclear is how separate instantiations of (1) for individual paradigms collectively account for the grammar as a whole. Because candidates in the /p/-, /t/-, and /s/-paradigms adopt different repair strategies and violate distinct faithfulness constraints, competition arises among the faithfulness constraints occupying the ranking space between $*\text{V-OBS-V}_{\text{Alter}}$ and $*\text{V-OBS-V}_{\text{Non-Alter}}$.

In the /p/-paradigm, the relevant faithfulness constraint violated by repair to [w] is IDENT(CONSONANTAL); in the /t/-paradigm, repair to [ɾ] violates IDENT(LATERAL); and in the /s/-paradigm, deletion of the obstruent segment violates MAX-C. Consequently, in each paradigm, the faithfulness constraint implicated in the repair must be outranked by the competing faithfulness constraints associated with the other two paradigms. To capture these cross-repair ranking relations, the ranking scheme in (1) must be expanded as in (2).

$$\begin{aligned} &*\text{V-OBS-V}_{\text{P-Alter}} \gg \text{IDENT(LATERAL)}, \text{MAX-C} \gg \\ &\text{IDENT(CONSONANTAL)} \gg *\text{V-OBS-V}_{\text{P-Non-Alter}} \\ &*\text{V-OBS-V}_{\text{T-Alter}} \gg \text{IDENT(CONSONANTAL)}, \text{MAX-C} \gg \\ &\text{IDENT(LATERAL)} \gg *\text{V-OBS-V}_{\text{T-Non-Alter}} \\ &*\text{V-OBS-V}_{\text{S-Alter}} \gg \text{IDENT(CONSONANTAL)}, \\ &\text{IDENT(LATERAL)} \gg \text{MAX-C} \gg *\text{V-OBS-V}_{\text{S-Non-Alter}} \end{aligned} \quad (2)$$

Undesirably, the expanded ranking requires the learner to maintain paradigm-specific instantiations of the grammar in addition to indexed constraints. Consequently, the learner must track both paradigm-specific rankings and the application domains of cloned constraints, thereby inheriting the learning burdens of both lexical indexation and the cophonology approach.

[3]When /s/-final stems are followed by consonants, the coda surfaces as [t] due to an independent coda neutralization rule, mapping all obstruents to their homorganic lenis stop in syllable-final position (Jun, 2010). This rule is not relevant to the alternation under investigation and is therefore set aside in the present analysis.

3.2 Lexically Scaled MaxEnt

To address this challenge in a more tractable setting, the present study adopts a probabilistic implementation of indexation—the lexically scaled MaxEnt model of Linzen et al. (2013); Hughto et al. (2019)—which captures paradigm-specific patterns without positing separate ranking systems. The model is a lexicalized variant of Maximum Entropy (MaxEnt) grammar (Goldwater and Johnson, 2003), designed to account for languages exhibiting both variability and lexical exceptionality. Non-lexically-specific variability is captured through general constraint weights, whereas lexically idiosyncratic patterns are modeled through morpheme-specific scaling factors incorporated into the harmony score used to compute MaxEnt probabilities.

$$\mathcal{H}_i = \sum_{\gamma \in \Gamma} \left(w_\gamma + \sum_{m \in \mu_i} s_{\gamma m} \right) v_{\gamma i} \tag{3}$$

In (3), morpheme-specific scaling factors $s_{\gamma m}$ are added to the general constraint weight w_γ before being multiplied by candidate i's violation profile, $v_{\gamma i}$, over the set of constraints $\gamma \in \Gamma$. As indicated by the subscripts, the general weight w_γ applies grammar-wide, whereas the scaling factors $s_{\gamma m}$ are specific to the individual morphemes m that constitute a given candidate. This additive penalty is conceptually analogous to lexical indexation: a higher-ranked class-specific markedness constraint (e.g., *V-OBS-V$_{\text{Alter}}$ in (1)) can be recast as an increased scaling factor applying to that lexical class. To illustrate this correspondence, consider the idealized learning outcome for the present phenomena shown in Table 3.

	*V-OBS-V	MAX	ID-LAT	ID-C
General weights	4	6	6	6
p-alternating	6	3	2	0
p-non-alternating	0	1	1	1
t-alternating	6	3	0	4
t-non-alternating	0	1	1	1
s-alternating	6	0	3	3
s-non-alternating	0	1	1	1
suffixes	0	0	0	0

Table 3: Hypothetical Learning Result

The rows beneath the general weights indicate scaling factors associated with each morpheme class, from which three learning predictions follow. First, *V-OBS-V must outrank the faithfulness constraints in alternating classes but not in non-alternating classes. Accordingly, only alternating classes bear a positive scaling weight of 6 on *V-OBS-V. As larger weights translate into higher rankings, alternating classes achieve the ordering *V-OBS-V$_{\text{Alter}}$ $\gg$ FAITH, whereas non-alternating classes, receiving no scaling, obey the reverse ordering FAITH $\gg$ *V-OBS-V$_{\text{Non-Alter}}$. These two partial orders constitute (1).

Second, for the correct repair strategy to emerge within each paradigm, the faithfulness constraint violated by the paradigm-specific alternation must receive the lowest scaling weight: ID-C for the /p/-alternation, ID-LAT for the /t/-alternation, and MAX for the /s/-alternation[4]. In Table 3, this is implemented by assigning a scaling weight of 0 to the violated faithfulness constraint in each alternating class, producing the effect of cross-repair competition within the paradigm.

Third, suffixes that function as *environmental triggers* ideally bear no scaling weights, since they are not themselves the locus of exceptionality.

When the learned grammar evaluates each conjugated item, the scaling factors associated with its constituent morphemes combine with the general weights to determine the effective constraint ranking. A non-lenited realization /ka.kap-ʌ/ in a p-alternating class (Table 1), for example, would be calculated as $4 + 6 + 0$, consisting of the general weight, stem-scale, and suffix-scale. In this way, distinctions along two dimensions—paradigm and exceptionality—are captured while avoiding the ranking complexity posited by the expanded indexation approach in (2).

Among the four hypothetical languages explored in Hughto et al. (2019), the *lexical language* most closely parallels the present phenomenon. In that language, lexically indexed prefixes categorically trigger vowel deletion, analogous to how different stem classes categorically opt for different strategies. In their learned grammar, deletion-triggering and deletion-blocking prefix classes receive complementary scaling on ALIGN and *CCC, such that the violated constraint in each class receives the smaller scaling weight (Table 4). This mirrors the first and second predictions proposed here, where non-alternating classes receive no scaling on *V-OBS-V, while the constraint violated by a paradigm-specific repair receives the lowest scaling within each alternating class. Moreover, because

[4]For notational brevity, this modeling work abbreviates IDENT(CONSONANTAL), IDENT(LATERAL), and MAX-C as ID-C, ID-LAT, and MAX.

stems are not the locus of lexical specificity, they receive no scaling, paralleling the present analysis in which suffixes ideally bear no scaling weights.

	*CCC	MAX	ALIGN
General Weights	4.6	0.0	4.1
Deleting Prefix	0.0	0.0	6.4
Non-Deleting Prefixes	5.3	0.0	0.0
Stems	0.0	0.0	0.0

Table 4: Learned weights of the *lexical language* (Hughto et al., 2019)

4 Experiments

4.1 Toy data Generation

Table 5 reproduces the statistics on Korean obstruent-final stems from the Sejong Corpus reported in Lee (2013). The Sejong Corpus, compiled by the National Institute of Korean Language (NIKL) as part of the 21st Century Sejong Project, has been widely used in previous studies (Albright and Kang, 2009; Jun, 2010), making it a well-established source for such statistics.

p-alternating	68 (72.34%)
p-non-alternating	5 (5.32%)
t-alternating	5 (5.32%)
t-non-alternating	7 (7.45%)
s-alternating	5 (5.32%)
s-non-alternating	4 (4.26%)

Table 5: Distributions of /p/-, /t/-, and /s/-final stems in the Sejong Corpus

Following the type counts in Table 5, we generated pseudo-stems with a CV-/p,s,t/ structure, reflecting Korean syllable phonotactics that restrict consonant clusters. Using the Romanization system provided by NIKL[5], onset consonants and vowels were drawn from the Romanized inventory of Korean segments. Onsets were selected from {g, kk, k, d, tt, t, b, pp, p, j, jj, ch, s, ss, h, n, m, r}, while vowels were restricted to the set of monophthongs {a, eo, o, u, eu, i, ae, e, oe, wi}. Random sampling from these sets yielded 180 unique CV sequences. These sequences were assigned without overlap to the six conjugation classes listed above and instantiated with the class-appropriate stem-final consonant (/p/, /t/, or /s/). The full list of toy stems is provided in Table 12 in Appendix A.

Because alternation is triggered in an intervocalic context, the inventory of novel suffixes was balanced with respect to onset presence: suffixes beginning with a consonant (–CV endings) and those beginning with a vowel (–V endings) were equally represented. The initial inventory consisted of two minimal endings, *–ta* and *–a*. It was then incrementally expanded by introducing additional vowel-initial and /t/-initial pairs, with vowels drawn from the same monophthong set used in stem generation. Table 13 in Appendix A summarizes the suffix sets across experimental conditions.

4.2 Distributional Conditions

Table 5 shows that /p/-alternating stems constitute the overwhelming majority of the data (72.34%). To isolate the effect of this distributional skew on the learner, we first conducted experiments under a simplified setting that excluded the /p/-paradigm.

The modeling data were initially restricted to /t/- and /s/-final stem conjugations, which we refer to as the *balanced condition*. In this setting, we examine two repair strategies for resolving *V-OBS-V: deletion and acquisition of the [+lateral] feature. Accordingly, triplets of UR–SR mappings are considered for each input form, corresponding to (a) the faithful candidate, (b) lenition via violation of IDENT(LATERAL), and (c) lenition via violation of MAX-C. For example, a t-alternating stem /kjɛ.dat/ in Table 1 would yield the candidates (a) [kjɛ.dat-a], (b) [kjɛ.dal-a], and (c) [kjɛ.da-a]. Table 14 in Appendix B presents the UR–SR mappings for the four stem classes under the two suffix conditions introduced above (–V and –CV endings).

Subsequently, we extended the model to the full paradigm by incorporating the /p/-ending data, creating the *imbalanced condition*. In this setting, loss of the [+consonantal] feature becomes an additional repair strategy alongside deletion and acquisition of [+lateral]. Accordingly, UR–SR mappings are represented as quadruplets for each input form, corresponding to (a) the faithful candidate, (b) lenition via violation of IDENT(CONSONANTAL)[6], (c) lenition via violation of IDENT(LATERAL), and (d) lenition via violation of MAX-C. Extending the previous example, /kjɛ.dat/ would now additionally yield (b) [kjɛ.daj-a] in addition to the previous three candidates. Table 15 in Appendix B presents a subset of the input tableaux under the two suffix

[5] https://www.korean.go.kr/front_eng/roman/roman_01.do

[6] When constructing candidate (b) for the /t/- and /s/-paradigms, the segmental alternation was instantiated as [j], a glide that preserves the place of articulation, similar to how /p/ surfaces as [w] while preserving labiality.

conditions.

4.3 Model Parameters

Learner settings followed the default implementation in Hughto et al. (2019). We used an L1 prior ($\lambda = 1.0$) and optimized the model using gradient descent with clipping (Tsuruoka et al., 2009). Weights and scales were initialized to 0.0, and training proceeded for 20,000 epochs with a learning rate of 0.001. All parameters were held constant across experimental conditions.

5 Results

5.1 Learning with the /t/- and /s/- paradigms

When only /t/- and /s/-final stems were included, type frequencies were roughly balanced both across paradigms and within each paradigm (Table 5). Nevertheless, the attested forms exhibit a 1:3 alternation ratio, because intervocalic lenition occurs only when alternating stems are followed by vowel-initial suffixes; in the other three configurations—alternating stem + CV suffix, non-alternating stem + V suffix, and non-alternating stem + CV suffix—faithful realization prevails. Given this overall distribution, the learner assigned less general weight to *V-OBS-V than to IDENT(LATERAL) and MAX-C.

	*V-OBS-V	MAX	ID-LAT
General weights	0	0.79	0.79
t-alternating stems	0	0	0
t-non-alternating stems	0	0	0
s-alternating stems	0	0	0
s-non-alternating stems	0	0	0
-V suffix	0	0	0
-CV suffix	0	2.16	2.16

Table 6: Learned weights (/t, s/ stems; two suffixes)

As such, while the majority pattern—faithful surfacing—is captured through general weights, the alternations are handled through lexical scaling. This division reflects the model's *least-cost* principle, which prioritizes general mechanisms for capturing dominant patterns while reserving lexical mechanisms for more restricted ones. The same pressure toward economy also shapes the learning of scaling factors. Jarosz (2025) shows that lexicalization tends to be stronger for higher-frequency items than for lower-frequency ones, and this tendency manifests in scaling weights initially being misattributed to the non-idiosyncratic suffixes.

When only two suffix types are available (Table 6), each stem occurs only twice in the attested forms—once with *-ta* and once with *-a*—while these two suffixes combine with all stem types. Consequently, when frequencies are evaluated per morpheme, suffixes occur more often than stems and thus emerge as the more general morpheme selected to account for exceptionality.

The same frequency bias persists within the suffix inventory. Because intervocalic lenition occurs in only one-fourth of the training data, the learner favors the -CV suffix, which creates a non-intervocalic environment, over the -V suffix. As a result, the highly frequent faithful realizations associated with the -CV suffix are preferentially captured through lexical scaling on MAX and ID-LAT rather than through scaling on *V-OBS-V in the intervocalic (-V suffix) environment.

	*V-OBS-V	MAX	ID-LAT
General weights	2.2	3.26	3.26
t-alternating stems	0.37	0	0
t-non-alternating stems	0	0	0
s-alternating stems	0.37	0	0
s-non-alternating stems	0	0	0
-V suffix	0	0	0
-CV suffix	0	0	0

Table 7: Learned weights (/t, s/ stems; four suffixes)

As the number of suffixes increases, stems combine with a larger set of suffixes and occur more frequently in the attested forms. This increases the likelihood that stem morphemes are lexicalized, reflected in the progressive redistribution of scale weights from suffixes to stems in Tables 7 and 8, and in Tables 16 and 17 in Appendix C.

The redistribution proceeds in stages. First, the learner captures the triggering of alternation by distinguishing alternating and non-alternating classes, assigning scale weight to *V-OBS-V in the former but not in the latter (Table 7). It then learns the paradigm-specific repair, establishing the ranking among faithfulness constraints that compete to derive class-specific forms. From Table 8 onward, the /t/- and /s/-alternating classes assign less scale weight to the violated constraint—ID-LAT and MAX, respectively—than to the competing faithfulness constraint, and this gap widens as additional suffix types are introduced.

This progression is expected under a frequency-sensitive learning mechanism. The /t/- and /s/-alternating classes have identical type frequencies (five types each), so the statistical signal for learn-

ing paradigm-specific behavior comes from the corresponding non-alternating classes. These differ only slightly in type frequency (seven for /t/-non-alternating and four for /s/-non-alternating), requiring more extensive exposure to attested forms before learners can acquire paradigm-specific distinctions beyond simply learning when alternation is triggered. In the present case, four suffix types are minimally required to distinguish alternating from non-alternating classes, and six suffix types to differentiate class-specific alternations.

	*V-OBS-V	MAX	ID-LAT
General weights	1.99	3.66	3.66
t-alternating stems	1.73	0.13	0
t-non-alternating stems	0	0	0
s-alternating stems	1.73	0	0.13
s-non-alternating stems	0	0	0
-V suffix	0	0	0
-CV suffix	0	0	0

Table 8: Learned weights (/t, s/ stems; six suffixes)

Improvements in the distribution of scale weights are reflected in the model's predictive performance. To evaluate this effect, we focus on alternating stems in intervocalic contexts, which represent the minority lenition-triggering portion of the training data and are thus harder for the learner to acquire. As shown in Fig. 1, accuracy in predicting the correct repair increases with the number of suffix types[7]. Notably, accuracy exceeds the 33% threshold only in the six-suffix condition[8], where the number of suffix types matches or exceeds the number of non-alternating stem types per paradigm (7 for the /t/-paradigm and 4 for the /s/-paradigm).

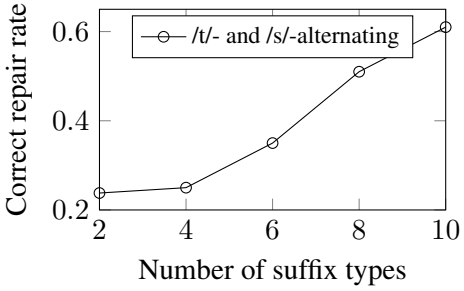


Figure 1: Repair accuracy of /t/- and /s/-alternating stems in intervocalic contexts

[7]The full numerical results for training accuracy are provided in Table 20 in Appendix D.

[8]Chance performance with three possible SR forms is approximately 33%.

5.2 Learning with the -/p/, -/t/, and -/s/ paradigms

Accommodating the /p/-final data creates an imbalance among the three alternating patterns. As shown in Table 5, the /p/-alternating class (72.34%) overwhelmingly outnumbers the /t/-alternating (5.32%) and /s/-alternating classes (5.32%). Although environments triggering intervocalic lenition remain outnumbered by those blocking it (approximately in a 1:3 ratio), most lenition tokens come from the /p/-alternating class, such that the number of glide-forming lenition tokens (68) approaches the frequency of faithful realizations (130).[9] Under these conditions, violation of ID-C emerges as the second default pattern after faithful realization. Consequently, when intervocalic lenition is triggered in the -V context, alternating stems favor repair via violation of ID-C.

This skewed distribution directly shapes the model's learning behavior. First, across all suffix conditions (Tables 9–11 and Tables 18 and 19 in Appendix C), the /p/-alternating class never acquires stem-scale weights. Second, ID-C, the constraint violated by this class, shows the greatest resistance to receiving general weight. In the initial two-suffix condition, all of its weight is borne by the *environmental trigger* -CV suffix. This bias persists as the suffix inventory increases and continues to shape the model's learning trajectory.

	*V-OBS-V	MAX	ID-LAT	ID-C
General Weights	0.69	2.6	2.6	0
p-alternating stems	0	0	0	0
p-non-alternating stems	0	0	0	0
t-alternating stems	0	0	0	0
t-non-alternating stems	0	0	0	0
s-alternating stems	0	0	0	0
s-non-alternating stems	0	0	0	0
-V suffix	0.69	0	0	0
-CV suffix	0	1.92	1.92	4.51

Table 9: Learned Weights (/p, t, s/ stems; two suffixes)

Excluding the effect of ID-C's resistance to receiving general weight, the progression from Tables 9–19 largely parallels the learning results obtained when the /p/ paradigm is excluded. With only two suffixes available, stems initially lack sufficient morphological evidence to learn their indexed status. The second stage—when the model begins to differentiate alternating from non-alternating classes (here limited to /t/ and /s/, since

[9]Comprised of all non-alternating stems in both suffix conditions ($(5 + 7 + 4) \times 2$) together with alternating stems in the non-triggering (-CV) environment ($68 + 5 + 5$).

/p/ no longer constitutes the minority pattern captured by lexical scale)—is delayed by the dominance of the /p/-alternations.

	*V-OBS-V	MAX	ID-LAT	ID-C
General Weights	4.02	5.21	5.21	2.14
p-alternating stems	0	0	0	0
p-non-alternating stems	0	0	0	1.41
t-alternating stems	0	0	0	1.41
t-non-alternating stems	0	0	0	1.41
s-alternating stems	0	0	0	1.41
s-non-alternating stems	0	0	0	1.41
-V suffix	0	0	0	0
-CV suffix	0	0	0	2.15

Table 10: Learned Weights (/p, t, s/ stems; four suffixes)

This is because alternation into a glide functions as a second default pattern alongside faithful surfacing, leaving the /t/- and /s/-alternating classes to compete with two majority patterns. Consequently, even in the four-suffix condition (Table 10), the model fails to differentiate among the six stem classes, as their stored vectors remain identical. Differentiation emerges only in Table 11, where scale weights begin to diverge between the /t/- and /s/-alternating classes.

	*V-OBS-V	MAX	ID-LAT	ID-C
General Weights	4.98	6.29	6.29	2.71
p-alternating stems	0	0	0	0
p-non-alternating stems	0	0	0	2.54
t-alternating stems	0.61	0	0	2.88
t-non-alternating stems	0	0	0	2.54
s-alternating stems	0.61	0	0	2.88
s-non-alternating stems	0	0	0	2.54
-V suffix	0	0	0	0
-CV suffix	0	0	0	1.53

Table 11: Learned Weights (/p, t, s/ stems; six suffixes)

The effect of statistical imbalance is further reflected in prediction accuracy across the alternating classes. As shown in Fig. 2, the curves for the /t/- and /s/-alternations fail to exceed the 25% chance-level threshold even in the six-suffix condition, unlike in the condition without /p/-alternation[10]. In contrast, the /p/-alternation maintains substantially higher accuracy throughout training. Nevertheless, the gap between the /p/-paradigm and the /t/- and /s/-paradigms gradually narrows as the number of suffix types increases, again highlighting the importance of suffix-type diversity.

[10]Chance-level accuracy in the imbalanced condition is 25%, since each UR corresponds to four possible SR candidates. See Table 21 in Appendix D for the exact numerical values.

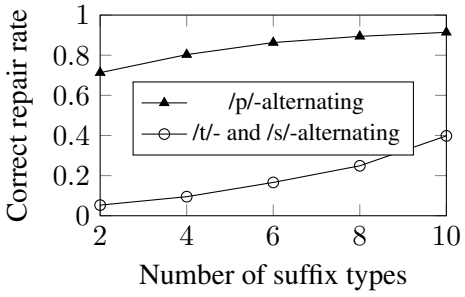


Figure 2: Repair accuracy of /p/-, /t/-, and /s/-alternating stems in intervocalic contexts

6 Discussion

6.1 Lexical statistics

One of the main findings is that, without a sufficiently large number of attested morphemic combinations, generalization cannot develop at the locus of exceptionality. This result highlights the statistical sensitivity of the model when learners are not exposed to enough morphological contexts in which exceptional items may occur.

The success of assigning scaling weights to the lexically idiosyncratic morphemes in the *lexical language* (see Table 4) can be interpreted in the same light. In Hughto et al. (2019), indexed items (prefixes) were outnumbered by non-indexed ones (stems), and the latter were replicated tenfold to achieve the reported performance. By contrast, our experiments involved the opposite asymmetry, where indexed items outnumbered non-indexed ones, and this unfavorable distribution was not compensated for through token-frequency amplification. Comparing the two results suggests that learning the locus of exceptionality critically depends on sufficient attestation in the input.

This observation raises the question of how suffixes are distributed in natural language. As non-indexed items, suffix statistics have received little attention and lack systematic documentation. We therefore conducted a corpus analysis of the Korean Google Treebank[11]. The corpus shows a distribution of stem types analogous to that in Table 5, while the suffix inventory far exceeds ten, comprising 29 -CV and 33 -V suffixes. Given such suffix type diversity in the empirical corpus data, absolute type frequency seems unlikely to pose a serious obstacle for real-world language acquisition.

[11]https://github.com/emorynlp/ud-korean/tree/master/google. The stem and suffix inventories identified in the corpus are reported in Tables 23 and 24 in Appendix E

6.2 Empirical Simulation

Nevertheless, an additional finding from our corpus analysis was that, although the number of available suffix types is sufficiently large, stems are not attested with every suffix type. In our earlier idealized learning scenario, full paradigm saturation was assumed, whereby every stem combines with all possible suffixes (Chan, 2008; Kodner, 2022). This assumption poorly reflects the sparsity of naturalistic input data, which typically follows a heavily skewed Zipfian distribution: a small number of stems occur across many derivational forms, whereas most combine with only a few suffixes.

This mismatch raises the possibility that learning difficulty may arise not from absolute type frequency, but from limited paradigm saturation. To test this hypothesis, we conducted an additional simulation using conjugation patterns extracted from the corpus. The results revealed the effects of Zipfian skewness: despite the availability of sufficient suffix types, stems still struggled to acquire lexical scaling, with only a small subset receiving nonzero scaling weights and even fewer learning sizable values.

By contrast, sizable scaling factors were widely assigned to suffixes, likely reflecting their broader attestation across stems. While most stems are attested with only a few suffixes, suffixes themselves appear with many different stems. This situation parallels the earlier two-suffix condition, in which stems were attested only twice; here, however, the challenge arises not from a limited number of morphemic counterparts, but from limited paradigm saturation across stem–suffix combinations.

Taken together, these results indicate that, under naturalistic input conditions, learning difficulty lies not in the absolute number of available types, but in the asymmetric degree of paradigm saturation between stems and suffixes. Furthermore, the Google Treebank UD corpus used in the empirical simulation has been identified as a suitable non-CDS proxy for acquisition input in computational learning systems (Kodner, 2022), supporting the ecological validity of the present findings.

6.3 Language acquisition

While analytically unexpected, the initial two-suffix condition—where lexical idiosyncrasy is attributed to suffixes—may be interpreted as a stage of overregularization. Because suffixes provide contextual information and function as phonologically regular *environmental triggers*, learners may initially assign excessive weight to their behavior, associating alternation with a general phonological rule before learning the locus of exceptionality itself. In particular, lenition is initially conditioned by suffixal phonological context rather than by stem-specific lexical exceptionality: it occurs before vowel-initial suffixes but not consonant-initial suffixes (Tables 6 and 9).

Such an interpretation may further address the critical discrepancy between model behavior and human language acquisition identified in Jarosz (2025), wherein regularization effects are strongest at early stages of learning in humans but are predicted to consistently strengthen over the course of training in Scaled MaxEnt.[12] Increasing the suffix inventory approximates a learning scenario in which the learner is exposed to progressively richer morphological exponence. Under this interpretation, the two-suffix condition corresponds to an early stage in the model's learning trajectory, where alternation is generalized as a phonologically conditioned pattern before being refined into stem-specific lexical exceptionality. As observed throughout Tables 7, 8, 16, and 17, and Tables 10, 11, 18, and 19, increasing the suffix inventory progressively reduces this tendency toward regularization, suggesting that the model's behavior may not entirely diverge from developmental patterns observed in human acquisition.

Nevertheless, two gaps remain if the model is to fully account for human learning of paradigm-specific exceptionality. First, it must address how appropriate scaling factors can be learned under conditions of limited paradigm saturation across stem–suffix combinations. Second, it must account for how statistically non-dominant repair strategies—such as those in the /t/ and /s/ paradigms—can be learned correctly in the presence of a dominant pattern.

References

Adam Albright and Yoonjung Kang. 2009. Predicting innovative alternations in korean verb paradigms. *Current issues in unity and diversity of languages: Collection of the papers selected from the CIL*, 18:1–20.

[12] Their time-course analysis tracks this tendency within a single training setting, whereas our comparison spans different suffix conditions. Nevertheless, because all simulations use batch learning and run for the same number of epochs, the results remain broadly comparable.

Arto Anttila. 2002. Morphologically conditioned phonological alternations. *Natural Language & Linguistic Theory*, 20(1):1–42.

Erwin Chan. 2008. *Structures and distributions in morphology learning*. University of Pennsylvania.

Andries W Coetzee. 2009. Learning lexical indexation. *Phonology*, 26(1):109–145.

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.

Coral Hughto, Andrew Lamont, Brandon Prickett, and Gaja Jarosz. 2019. Learning exceptionality and variation with lexically scaled maxent. In *Proceedings of the Society for Computation in Linguistics (SCIL) 2019*, pages 91–101.

Sharon Inkelas, C Orhan Orgun, and Cheryl Zoll. 1997. The implications of lexical exceptions for the nature of grammar. *Derivations and constraints in phonology*, pages 393–418.

Sharon Inkelas and Cheryl Zoll. 2003. Is grammar dependence real. *Ms, UC Berkeley and MIT. ROA-587*.

Gaja Jarosz. 2025. Incremental learning of lexically specific morphophonology: an integrative approach. *Linguistics Vanguard*, (0).

Jongho Jun. 2010. Stem-final obstruent variation in korean. *Journal of East Asian Linguistics*, 19(2):137–179.

Jordan Kodner. 2022. Computational models of morphological learning. In *Oxford Research Encyclopedia of Linguistics*.

Ponghyung Lee. 2013. Constraint cloning and lexical listing in korean irregular verbs and adjectives. *Studies in Linguistics*.

Sechang Lee. 1996. On the irregular verbs in korean. In *Proceedings of the 11th Pacific Asia Conference on Language, Information and Computation*, pages 469–478.

Tal Linzen, Sofya Kasyanenko, and Maria Gouskova. 2013. Lexical and phonological variation in russian prepositions. *Phonology*, 30(3):453–515.

Joe Pater. 1999. Austronesian nasal substitution and other nc effects. *The prosody-morphology interface*, pages 310–343.

Joe Pater. 2000. Non-uniformity in english secondary stress: the role of ranked and lexically specific constraints. *Phonology*, 17(2):237–274.

Joe Pater. 2009. Morpheme-specific phonology: Constraint indexation and inconsistency resolution.

Sang-Jik Rhee. 1996. The identities of the stem-final consonants of p-and t-irregular verbs in korean. In *Proceedings of the KSPS conference*, pages 476–481. The Korean Society Of Phonetic Sciences And Speech Technology.

Bruce Tesar and Alan Prince. 2003. Using phonotactics to learn phonological alternations. In *Proceedings from the annual meeting of the Chicago Linguistic Society*, volume 39, pages 241–269. Chicago Linguistic Society.

Bruce Tesar and Paul Smolensky. 1998. Learnability in optimality theory. *Linguistic inquiry*, 29(2):229–268.

Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou. 2009. Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 477–485.

A Appendix

Category	Stems
p-alternating	neop- ppaep- sep- ssep- gop- bwip- seup- dep- dop- baep- nep- kkeup- moep- kkip- kkop- ssaep- chap- rep- kop- teup- cheop- tep- heup- rwip- jjwip- paep- bop- bup- mip- dwip- choep- sop- swip- pwip- jjeop- keop- soep- ssap- taep- mwip- rup- chwip- bep- peup- ppep- chop- keup- geup- nip- ppip- saep- noep- kkap- jjup- gwip- jjep- meup- ttop- jep- daep- joep- mop- mep- ssoep- ttaep- reop- jjop- ppap-
p-non-alternating	jaep- kkep- jjip- rap- toep-
t-alternating	tat- jat- ket- pot- chit-
t-non-alternating	ppeut- goet- jot- pput- nat- kkut- sit-
s-alternating	ges- kkaes- hoes- koes- nos-
s-non-alternating	poes- meos- jjaes- ches-

Table 12: Toy stems

Condition	Suffix inventory
2-suffix	*-ta*, *-a*
4-suffix	*-ta*, *-a*, *-te*, *-e*
6-suffix	*-ta*, *-a*, *-te*, *-e*, *-tu*, *-u*
8-suffix	*-ta*, *-a*, *-te*, *-e*, *-tu*, *-u*, *-to*, *-o*
10-suffix	*-ta*, *-a*, *-te*, *-e*, *-tu*, *-u*, *-to*, *-o*, *-ti*, *-i*

Table 13: Toy suffixes

B Appendix

UR	Candidate	% occurrences	*V-OBS-V	MAX	ID-LAT
/tat-a$_{\text{T-Alter}}$/	(a) [tata]	0.0	1	0	0
	(b) [tara]	1.0	0	0	1
	(c) [taa]	0.0	0	1	0
/tat-ta$_{\text{T-Alter}}$/	(a) [tatta]	1.0	0	0	0
	(b) [tarta]	0.0	0	0	1
	(c) [tata]	0.0	0	1	0
/jot-a$_{\text{T-Non-Alter}}$/	(a) [jota]	1.0	1	0	0
	(b) [jora]	0.0	0	0	1
	(c) [joa]	0.0	0	1	0
/jot-ta$_{\text{T-Non-Alter}}$/	(a) [jotta]	1.0	0	0	0
	(b) [jorta]	0.0	0	0	1
	(c) [jota]	0.0	0	1	0
/ges-a$_{\text{S-Alter}}$/	(a) [gesa]	0.0	1	0	0
	(b) [gera]	0.0	0	0	1
	(c) [gea]	1.0	0	1	0
/ges-ta$_{\text{S-Alter}}$/	(a) [gesta]	1.0	0	0	0
	(b) [gerta]	0.0	0	0	1
	(c) [geta]	0.0	0	1	0
/poes-a$_{\text{S-Non-Alter}}$/	(a) [poesa]	1.0	1	0	0
	(b) [poera]	0.0	0	0	1
	(c) [poea]	0.0	0	1	0
/poes-ta$_{\text{S-Non-Alter}}$/	(a) [poesta]	1.0	0	0	0
	(b) [poerta]	0.0	0	0	1
	(c) [poeta]	0.0	0	1	0

Table 14: Input data tableau for a statistically-balanced condition

UR	Candidate	% occurrences	*V-OBS-V	MAX	ID-C	ID-LAT
/neop-a$_{\text{P-Alter}}$/	(a) [neopa]	0.0	1	0	0	0
	(b) [neowa]	1.0	0	0	1	0
	(c) [neola]	0.0	0	0	0	1
	(d) [neoa]	0.0	0	1	0	0
/neop-ta$_{\text{P-Alter}}$/	(a) [neopta]	1.0	1	0	0	0
	(b) [neowta]	0.0	0	0	1	0
	(c) [neolta]	0.0	0	0	0	1
	(d) [neota]	0.0	0	1	0	0
/jaep-a$_{\text{P-Non-Alter}}$/	(a) [jaepa]	1.0	1	0	0	0
	(b) [jaewa]	0.0	0	0	1	0
	(c) [jaela]	0.0	0	0	0	1
	(d) [jaea]	0.0	0	1	0	0
/jaep-ta$_{\text{P-Non-Alter}}$/	(a) [jaepta]	1.0	1	0	0	0
	(b) [jaewta]	0.0	0	0	1	0
	(c) [jaelta]	0.0	0	0	0	1
	(d) [jaeta]	0.0	0	1	0	0
/tat-a$_{\text{T-Alter}}$/	(a) [tata]	0.0	1	0	0	0
	(b) [taja]	0.0	0	0	1	0
	(c) [tara]	1.0	0	0	0	1
	(d) [taa]	0.0	0	1	0	0
/tat-ta$_{\text{T-Alter}}$/	(a) [tatta]	1.0	0	0	0	0
	(b) [tajta]	0.0	0	0	1	0
	(c) [tarta]	0.0	0	0	0	1
	(d) [tata]	0.0	0	1	0	0
/jot-a$_{\text{T-Non-Alter}}$/	(a) [jota]	1.0	1	0	0	0
	(b) [joja]	0.0	0	0	1	0
	(c) [jora]	0.0	0	0	0	1
	(d) [joa]	0.0	0	1	0	0
/jot-ta$_{\text{T-Non-Alter}}$/	(a) [jotta]	1.0	0	0	0	0
	(b) [jojta]	0.0	0	0	1	0
	(c) [jorta]	0.0	0	0	0	1
	(d) [jota]	0.0	0	1	0	0
/ges-a$_{\text{S-Alter}}$/	(a) [gesa]	0.0	1	0	0	0
	(b) [geja]	0.0	0	0	1	0
	(c) [gera]	0.0	0	0	0	1
	(d) [gea]	1.0	0	1	0	0
/ges-ta$_{\text{S-Alter}}$/	(a) [gesta]	1.0	0	0	0	0
	(b) [gejta]	0.0	0	0	1	0
	(c) [gerta]	0.0	0	0	0	1
	(d) [geta]	0.0	0	1	0	0
/poes-a$_{\text{S-Non-Alter}}$/	(a) [poesa]	1.0	1	0	0	0
	(b) [poeja]	0.0	0	0	1	0
	(c) [poera]	0.0	0	0	0	1
	(d) [poea]	0.0	0	1	0	0
/poes-ta$_{\text{S-Non-Alter}}$/	(a) [poesta]	1.0	0	0	0	0
	(b) [poejta]	0.0	0	0	1	0
	(c) [poerta]	0.0	0	0	0	1
	(d) [poeta]	0.0	0	1	0	0

Table 15: Input data tableau for statistically imbalanced condition

C Appendix

	*V-OBS-V	MAX	ID-LAT
General weights	1.80	3.85	3.85
t-alternating stems	2.76	0.75	0
t-non-alternating stems	0	0	0
s-alternating stems	2.76	0	0.75
s-non-alternating stems	0	0	0
-V suffix	0	0	0
-CV suffix	0	0	0

Table 16: Learned weights (/t, s/ stems; eight suffixes)

	*V-OBS-V	MAX	ID-LAT
General weights	1.72	4.04	4.04
t-alternating stems	3.43	1.14	0
t-non-alternating stems	0	0	0
s-alternating stems	3.43	0	1.14
s-non-alternating stems	0	0	0
-V suffix	0	0	0
-CV suffix	0	0	0

Table 17: Learned weights (/t, s/ stems; ten suffixes)

	*V-OBS-V	MAX	ID-LAT	ID-C
General Weights	6.14	7.76	7.76	3.67
p-alternating stems	0	0	0	0
p-non-alternating stems	0	0	0	3.24
t-alternating stems	1.62	0.0005	0	4.09
t-non-alternating stems	0	0	0	3.24
s-alternating stems	1.62	0	0.0005	4.09
s-non-alternating stems	0	0	0	3.24
-V suffix	0	0	0	0
-CV suffix	0	0	0	0.58

Table 18: Learned Weights (/p, t, s/ stems; eight suffixes)

	*V-OBS-V	MAX	ID-LAT	ID-C
General Weights	7.11	8.98	8.98	4.46
p-alternating stems	0	0	0	0
p-non-alternating stems	0	0	0	3.76
t-alternating stems	2.55	0.69	0	5.2
t-non-alternating stems	0	0	0	3.76
s-alternating stems	2.55	0	0.69	5.2
s-non-alternating stems	0	0	0	3.76
-V suffix	0	0	0	0
-CV suffix	0	0	0	0

Table 19: Learned Weights (/p, t, s/ stems; ten suffixes)

D Appendix

Two suffixes

	t-conjugation				s-conjugation			
	t-alternating		t-non-alternating		s-alternating		s-non-alternating	
	-V	-CV	-V	-CV	-V	-CV	-V	-CV
P(faithful)	0.523	0.904	0.523	0.904	0.523	0.904	0.523	0.904
P(lateralization)	**0.238**	0.047	0.238	0.047	0.238	0.047	0.238	0.047
P(deletion)	0.238	0.047	0.238	0.047	**0.238**	0.047	0.238	0.047

Four suffixes

	t-conjugation				s-conjugation			
	t-alternating		t-non-alternating		s-alternating		s-non-alternating	
	-V	-CV	-V	-CV	-V	-CV	-V	-CV
P(faithful)	0.50	0.93	0.59	0.93	0.50	0.93	0.59	0.93
P(lateralization)	**0.25**	0.03	0.20	0.03	0.25	0.03	0.20	0.03
P(deletion)	0.25	0.03	0.20	0.03	**0.25**	0.03	0.20	0.03

Six suffixes

	t-conjugation				s-conjugation			
	t-alternating		t-non-alternating		s-alternating		s-non-alternating	
	-V	-CV	-V	-CV	-V	-CV	-V	-CV
P(faithful)	0.33	0.95	0.73	0.95	0.33	0.95	0.73	0.95
P(lateralization)	**0.35**	0.024	0.14	0.03	0.31	0.024	0.14	0.02
P(deletion)	0.31	0.021	0.14	0.021	**0.35**	0.024	0.14	0.02

Eight suffixes

	t-conjugation				s-conjugation			
	t-alternating		t-non-alternating		s-alternating		s-non-alternating	
	-V	-CV	-V	-CV	-V	-CV	-V	-CV
P(faithful)	0.25	0.97	0.79	0.96	0.25	0.97	0.79	0.96
P(lateralization)	**0.51**	0.10	0.21	0.02	0.24	0.01	0.10	0.02
P(deletion)	0.24	0.01	0.09	0.02	**0.51**	0.02	0.10	0.02

Ten suffixes

	t-conjugation				s-conjugation			
	t-alternating		t-non-alternating		s-alternating		s-non-alternating	
	-V	-CV	-V	-CV	-V	-CV	-V	-CV
P(faithful)	0.20	0.97	0.84	0.96	0.20	0.987	0.84	0.96
P(lateralization)	**0.61**	0.02	0.08	0.02	0.19	0.003	0.08	0.02
P(deletion)	0.19	0.01	0.08	0.02	**0.61**	0.01	0.08	0.02

Table 20: Training-data P(E) under the balanced condition

Two suffixes

	p-conjugation				t-conjugation				s-conjugation			
	p-alternating		p-non-alternating		t-alternating		t-non-alternating		s-alternating		s-non-alternating	
	-V	-CV	-V	-CV	-V	-CV	-V	-CV	-V	-CV	-V	-CV
P(faithful)	0.181	0.968	0.181	0.968	0.181	0.968	0.181	0.968	0.181	0.968	0.181	0.968
P(glide)	**0.713**	0.011	0.713	0.011	0.713	0.011	0.713	0.011	0.713	0.011	0.713	0.011
P(lateralization)	0.053	0.011	0.053	0.011	**0.053**	0.011	0.053	0.011	0.053	0.011	0.053	0.011
P(deletion)	0.053	0.011	0.053	0.011	0.053	0.011	0.053	0.011	**0.053**	0.011	0.053	0.011

Four suffixes

	p-conjugation				t-conjugation				s-conjugation			
	p-alternating		p-non-alternating		t-alternating		t-non-alternating		s-alternating		s-non-alternating	
	-V	-CV	-V	-CV	-V	-CV	-V	-CV	-V	-CV	-V	-CV
P(faithful)	0.123	0.976	0.313	0.986	0.313	0.986	0.313	0.986	0.313	0.986	0.313	0.986
P(glide)	**0.803**	0.013	0.497	0.003	0.497	0.003	0.497	0.003	0.497	0.003	0.497	0.003
P(lateralization)	0.037	0.005	0.095	0.005	**0.095**	0.005	0.095	0.005	0.095	0.005	0.095	0.005
P(deletion)	0.037	0.005	0.095	0.005	0.095	0.005	0.095	0.005	**0.095**	0.005	0.095	0.005

Six suffixes

	p-conjugation				t-conjugation				s-conjugation			
	p-alternating		p-non-alternating		t-alternating		t-non-alternating		s-alternating		s-non-alternating	
	-V	-CV	-V	-CV	-V	-CV	-V	-CV	-V	-CV	-V	-CV
P(faithful)	0.089	0.982	0.434	0.995	0.334	0.995	0.434	0.995	0.334	0.995	0.434	0.995
P(glide)	**0.863**	0.014	0.332	0.001	0.333	0.001	0.332	0.001	0.333	0.001	0.332	0.001
P(lateralization)	0.024	0.002	0.117	0.002	**0.166**	0.002	0.117	0.002	0.166	0.002	0.117	0.002
P(deletion)	0.024	0.002	0.117	0.002	0.166	0.002	0.117	0.002	**0.166**	0.002	0.117	0.002

Eight suffixes

	p-conjugation				t-conjugation				s-conjugation			
	p-alternating		p-non-alternating		t-alternating		t-non-alternating		s-alternating		s-non-alternating	
	-V	-CV	-V	-CV	-V	-CV	-V	-CV	-V	-CV	-V	-CV
P(faithful)	0.076	0.985	0.538	0.998	0.25	0.998	0.538	0.998	0.25	0.998	0.537	0.998
P(glide)	**0.894**	0.014	0.249	0.0005	0.25	0.0002	0.249	0.0005	0.25	0.0002	0.249	0.0005
P(lateralization)	0.015	0.0004	0.106	0.0004	**0.249**	0.0004	0.106	0.0004	0.249	0.0004	0.106	0.0004
P(deletion)	0.015	0.0004	0.106	0.0004	0.249	0.0004	0.106	0.0004	**0.249**	0.0004	0.106	0.0004

Ten suffixes

	p-conjugation				t-conjugation				s-conjugation			
	p-alternating		p-non-alternating		t-alternating		t-non-alternating		s-alternating		s-non-alternating	
	-V	-CV	-V	-CV	-V	-CV	-V	-CV	-V	-CV	-V	-CV
P(faithful)	0.064	0.988	0.61	0.999	0.2	0.999	0.61	0.999	0.2	0.999	0.611	0.999
P(glide)	**0.914**	0.011	0.20	0.0002	0.2	0	0.2	0	0.2	0	0.2	0
P(lateralization)	0.010	0.0001	0.094	0.0001	**0.398**	0	0.094	0	0.2	0	0.094	0
P(deletion)	0.010	0.0001	0.094	0.0001	0.2	0	0.09	0	**0.398**	0.0001	0.09	0

Table 21: Training-data P(E) under the imbalanced condition

E Appendix

p-alternating 45 (45%)
p-non-alternating 17 (17%)
t-alternating 8 (8%)
t-non-alternating 14 (14%)
s-alternating 8 (8%)
s-non-alternating 8 (8%)

Table 22: Distribution of /p/, /t/, and /s/-final stems in the Google Treebank Corpus

Category	Stems
p-alternating	gakkap- gomap- gop- gup- geurip- deop- dop- dukkeop- tteugeop- maep- mugeop- museop- budeureop- saerop- swip- sikkeureop- akkap- areumdap- aswip- antakkap- eoryeop- jeulgeop- chup- himgyeop- kkadarop- deoreop- maekkeureop- bangap- bukkeureop- sonswip- singgeop- useup- jeonggyeop- gabyeop- goerop- gwiyeop- nollap- nup- duryeop- duteop- maeseop- ansseureop- eodup- oerop- chagap-
p-non-alternating	dwijip- butjap- ppob- ssip- ip- jap- job- barojap- hwijip- kkojip- kkob- sarojap- sonjap- jip- jjip- charyeip- himip-
t-alternating	geot- deut- mut- sit- kkaedat- ilkkeut- git- but-
t-non-alternating	dat- mit- bat- ssot- eot- gut- mulryeobat- beorimbat- injeongbat- geot- got- tteut- mut- ieobat-
s-alternating	nas- bus- is- jis- maedeupjis- nongsajis- dwis- yeonis-
s-non-alternating	ssis- us- johas- chisos- beos- byas- ppaeas- is-

Table 23: /p/, /t/, and /s/-final stems in the Google Treebank Corpus (romanized)[a]

[a] Romanization follows https://www.korean.go.kr/front_eng/roman/roman_01.do.

Context	Suffix inventory
-CV	-geona -ge -get -go -goyo -guyo -gi -gillae -na -nayo -neyo -neureago -neun -neunda -neundago -neunde -da -daga -dago -daneun -deogunyo -deon -dorok -seumnida -ja -janeun -jamaja -ji -jiman
-V	-n -nde -l -m -myeo -myeon -myeonse -a -ado -aseo -aya -ayo -at -eo -eodo -eoseo -eoya -eoyo -eot -eotseot -euna -euni -eureo -eureogo -eureoneun -eumyeo -eumyeon -eumyeonse -euseyo -eusi -eun -eul -eum

Table 24: Suffix inventories in the Google Treebank Corpus (romanized)[a]

[a] Romanization follows https://www.korean.go.kr/front_eng/roman/roman_01.do.

Adaptive Speech Perception: Empirical Indeterminacy and a Path Forward

Shawn Cummings[1] **T. Florian Jaeger**[2] **Chigusa Kurumada**[2] **Xin Xie**[1]
[1]Department of Language Science, University of California, Irvine
[2]Department of Brain and Cognitive Sciences, University of Rochester
{scummin2,xxie14}@uci.edu
{fjaeger,ckuruma2}@ur.rochester.edu

Abstract

Human listeners rapidly adapt to unfamiliar talkers, but the underlying computational mechanisms remain contested. Three candidate hypotheses—pre-linguistic normalization, changes in phonetic category representations, and changing decision biases—have largely been pursued in separation, using subfield-specific paradigms. Researchers working in these paradigms often assume that adaptivity observed in their particular paradigm can only be explained by one of the three mechanisms. We test this assumption for one of the most popular experimental paradigms (lexically-guided perceptual learning or LGPL) using a unified computational framework (ASP). We apply ASP to the largest existing LGPL data: 89,600 categorization responses from over 1000 listeners after lexically-guided exposure to 32 different stimulus sets. Despite the unprecedented scale of these data, we find that behavioral data are equally compatible with all three candidate mechanisms. We discuss how model-guided stimulus selection can increase the diagnosticity of future LGPL experiments. Our simulation code can easily be adapted to other experimental paradigms.

1 Introduction

One of the core computational challenges facing the human brain is inducing stable, generalizable structure from noisy and variable perceptual signals. Speech perception is a particularly vivid illustration of this challenge. Even a simple phonological contrast like /s/ (e.g., *sip*) vs. /ʃ/ (e.g., *ship*) is realized across a vast acoustic space: a talker's /s/ may be acoustically more similar to another talker's /ʃ/ than to their own prior productions, due to variation in vocal tract anatomy, speaking rate, dialect and social identity, etc. Yet listeners perceive speech with remarkable consistency across talkers (for review, Johnson and Sjerps, 2021; Weatherholtz and Jaeger, 2016).

Decades of experimental work have established that this robustness is not merely passive tolerance of variability, but reflects active adaptivity. Listeners recalibrate their perception based on the acoustics of recent inputs, so that the same acoustic signal might be recognized as an instance of a different linguistic category (e.g., phonemes, syllables, or words). Unlike most commonly available automatic speech recognition systems, this adaptivity can be rapid—sometimes occurring within a single utterance—and yet can persist for days. This adaptivity is now considered a fundamental property of human speech perception, documented across phonetic contrasts, languages, and listener populations (for review, Bent and Baese-Berk, 2021; Johnson and Sjerps, 2021).

The central question we seek to address here concerns the computational mechanisms responsible for this adaptivity. It remains fundamentally unclear what computations support adaptive speech perception, how their consequences interact to jointly shape perception, and what determines the relative engagement of different mechanisms across tasks, stimuli, and contexts. Here, we contribute to recent demonstrations that **less is known about the mechanisms underlying speech perception than commonly assumed**, and begin to demonstrate how **minor changes to the stimulus design of a highly popular experimental paradigm could substantially enhance its diagnosticity** about the underlying mechanisms. We pursue these goals within a recently proposed minimal computational framework for adaptive speech perception (ASP), described below.

ASP (Xie et al., 2023) incorporates competing proposals about the mechanisms underlying adaptive speech perception, allowing us to simulate and analyze changes in perception under different hypotheses about these mechanisms. Through the computational simulations we present—fit against, and informed by, human behavioral data—we hope

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 513–528
July 3-4, 2026 ©2026 Association for Computational Linguistics

to re-emphasize an important insight that—while repeatedly and cogently made in previous work (Newell, 1973; Yarkoni and Westfall, 2017; Guest and Martin, 2021)—remains underappreciated among many experimenters: informally stated hypotheses can productively guide research in a field's early stages; however, there comes a time in the development of a field when progress requires sufficiently specific computational models of the remaining plausible accounts to effectively contrast them across paradigms and phenomena. This is particularly true when—as is typically the case for any non-trivial problem in the language sciences—multiple mechanisms contribute to observable behavior, so that research needs to go beyond qualitative existence proofs.

1.1 The problem: unknown uncertainties

Three broad classes of proposals have been advanced in the literature to account for adaptivity in speech perception (for review, Xie et al., 2023). The first attributes changes to **pre-linguistic signal normalization**: low-level auditory processes that track the overall statistical properties of the input—such as the mean of a cue distribution—and adjust perception accordingly, without reference to linguistic categories. The second attributes adaptation to **changes in linguistic category representations**: listeners update their implicit knowledge of the cue distributions associated with phonological categories (e.g., shifting or expanding the distribution for /s/ along its phonetic cues). The third attributes adaptation to **changes in decision-making**: listeners adjust response biases or decision criteria rather than the perceptual or representational processes that precede them. These three classes of proposals assume fundamentally different cognitive architectures, with divergent implications for theories of speech perception, language acquisition, and the malleability of neural representations of phonological categories.

Despite decades of research, principled comparison of these mechanisms are largely lacking. Theoretical proposals have often remained underspecified—formulated verbally rather than as computational models—and predictions from competing mechanisms are rarely tested against the same data (for review, Tan and Jaeger, 2025; Xie et al., 2023). As a result, different research communities continue to operate in parallel, each citing evidence from different paradigms in support of their preferred account. For instance, researchers who conceive of their experimental paradigm as tapping into normalization processes might point to findings that non-speech acoustic contexts can shift the perception of subsequent speech inputs (Huang and Holt, 2009; Laing et al., 2012)—a result that can be taken to challenge representational accounts. Researchers who attribute changes in listeners' behavior to changes in phonological category representations, on the other hand, might cite evidence that adaptation can be affected by category labels: the same input inferred to be an /s/ can have different consequences on subsequent perception than if that input is inferred to be an /f/ (Norris et al., 2003; Jesse, 2021)—a finding argued to be incompatible with normalization. Similarly, researchers favoring decision-making as the source of behavioral changes might point to neuroimaging evidence implicating prefrontal and parietal regions (Myers and Mesite, 2014)—typically taken to reflect post-perceptual processes rather than low-level signal transformation or representational change.

It is, however, inherently risky to argue based on qualitative findings and theoretical intuitions in the absence of a common theoretical vocabulary. For instance, the argument that signal normalization cannot account for the findings in Norris et al. (2003) rests on the assumption that normalization does not consider the category inferred to underlie the current input. While this assumption applies to some influential normalization accounts (e.g., C-CuRE, McMurray and Jongman 2011), alternative accounts that normalize phonetic cues based on inferred category labels have long existed (Nearey and Assmann, 2007), and have been shown to better explain speech perception (Barreda and Jaeger, 2025). This illustrates that even strong intuitions about the interpretation of existing results often rely on assumptions not shared by all researchers. An alternative approach—which we pursue here—is to develop models that deliver clear, quantitative predictions that can be evaluated using standard goodness-of-fit metrics for model comparison. This approach, at a minimium, makes all relevant assumptions explicit.

1.2 A path forward: a computational framework for adaptive speech perception (ASP)

As a step towards this goal, Xie et al. (2023) introduced a computational framework for adaptive speech perception (ASP). As illustrated in Figure 1, ASP formally specifies a categorization model

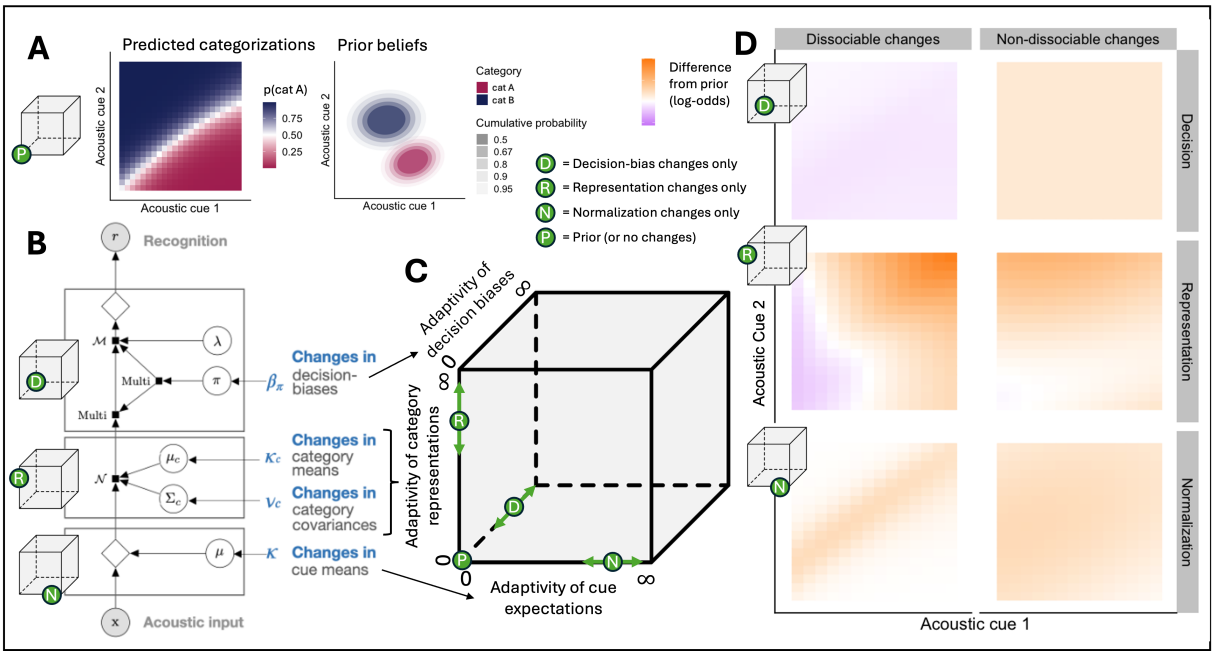


Figure 1: The ASP framework, parameterizations, and potential predictions. **Panel A:** One potential set of prior beliefs about cue-category mappings in a two-dimensional acoustic cue space, and resultant predictions from the categorization model. **Panel B:** graphic illustration of the categorization model and parameters over which the change models operate. **Panel C:** visualization of the multidimensional parameter space of ASP, with marked annotations for cases where only a single model contributes to adaptation and the case where no models exhibit any flexibility, resulting in no changes from prior beliefs. **Panel D:** Predicted differences between each model's prediction and a model without any adaptation (= predicted adaptive changes of categorization responses). Columns depict two possible scenarios in which these predictions diverge (left) or converge (right), depending on the acoustic characteristics of the exposure input, where convergence contributes to empirical indeterminacy.

that maps acoustic inputs to linguistic categories via normalization (linear transformations of the acoustic inputs), category representations (category likelihoods), and decision-making (category priors that represent decision and response biases). ASP also specifies change models for each of these three components to describe how normalization, category representations, and decision biases might be adapted in response to recent exposure.

ASP leaves open whether humans draw on only a single mechanism when adapting to unfamiliar speech. All three change models could operate at the same time. Figure 1 (panel C) illustrates this point by representing parameterization in ASP as a 3D space, where each axis determines a hypothetical listener's flexibility or willingness to change according to a different model. The 'true' characterization of a given listener could lie anywhere in this three dimensional space. Before the field can address this question productively, however, we need to address a simpler question: whether existing data can even distinguish between scenarios in which only one change model is operating. These scenarios represent how theories are often evoked in the field: in isolation and at the exclusion of one another. These scenarios also provide the strongest potential for measurable differences in predictions. If existing data cannot even distinguish between the extreme scenarios in which only one model is active, those data certainly cannot be used to determine what mixture of mechanisms listeners might draw on.

Given the separate computational architectures of each model, one might assume them to beget very different predictions (Figure 1D, left column). However, initial simulations that applied ASP to synthetic data produced a striking result: the signature findings from two highly popular and frequently cited experimental paradigms in the field (lexically-guided perceptual learning, henceforth LGPL, and nonnative accent adaptation) are qualitatively compatible with all three mechanisms (Xie et al., 2023) . This is illustrated in Figure 1D (right column).

This raises two questions, the answers to which will determine what a feasible path towards more diagnostic experimental paradigms will look like. First, is the empirical indeterminacy observed by

Xie and colleagues an artifact of the specific, simplified stimulus configurations used in the initial simulations? Perhaps the mechanisms generate divergent, distinguishable predictions for actual experiments with natural speech stimuli—especially once one considers more than a single experiment at a time. If so, popular paradigms might *in principle* risk being uninformative about the three mechanisms, while in practice providing sufficient signal about the (mixture of) mechanisms that underlie listeners' responses. Second, is the empirical indeterminacy an inherent artifact of the low-dimensional behavioral measures assumed in Xie and colleagues' simulations—specifically, the two-alternative forced choice (2AFC) task? 2AFC tasks continue to dominate the field, in part because they are often simple and efficient for both participants and experimenters. They have, however, long been critiqued for being less informative than other tasks (see also recent debate about 2AFC vs. visual analogue scale, Apfelbaum et al., 2022). It is thus possible that no amount of tweaking the stimulus selection will make 2AFC experiments diagnostic of mechanisms underlying adaptive speech perception.

Here, we focus on the first question and demonstrate that findings from one of the most popular paradigms in the field (LGPL, Norris et al., 2003; Kraljic and Samuel, 2005) do not, in fact, provide diagnostic evidence for one mechanism over the others—contrary to many researchers' intuitions. This matters beyond the immediate paradigm: LGPL findings have been used as a foundation for theoretical claims in adjacent subfields. For instance it has been claimed that older adults have decreased flexibility in the adjustment of phoneme categories, based on reduced magnitudes of boundary shifts in LGPL studies (Scharenborg and Janse 2013). Such claims presuppose that LGPL results can be attributed to a specific mechanism. We stress that our findings do not indicate that LGPL is inherently uninformative about mechanisms; as we illustrate in the Discussion, relatively modest changes to stimulus selection can substantially improve its diagnosticity. Moreover, the indeterminacy we identify is unlikely to be unique to LGPL: any paradigm whose theoretical interpretations rest on informally stated hypotheses rather than formalized computational models risks the same problem. In the Discussion, we turn to the second question and illustrate how ASP-guided stimulus selection practices hold promise to remedy the indeterminacy.

2 Data

We use the largest available database of LGPL experiments (Cummings, 2025): a total of 89,600 post-exposure categorization responses from 1,280 listeners across 16 LGPL experiments—each with two exposure conditions—that only differed in their exposure and test stimuli.

In LGPL experiments, listeners are exposed to speech in which a perceptually ambiguous sound (e.g., midway between /s/ and /ʃ/) is embedded in words that disambiguate its intended category (e.g., *per/?/onal* biases interpretation toward /s/). Their subsequent categorization of that sound on a test continuum shifts in the direction consistent with the lexical label, taken as a measure of adaptation to the talker's pronunciation. The design, procedure, and stimuli of the 16 experiments in Cummings (2025) closely follow conventions of the field, including the use of a lexical decision task during exposure and a 2AFC task during test ("Did the talker say *asi* or *ashi*?"). A schematic for this paradigm can be found in Appendix A1.

2.1 Stimuli

The **exposure** stimuli included auditory recordings of 100 English words and 100 nonwords (e.g., *baliber*) produced by 16 talkers (8 male and 8 female) of American English. 20% of the words contained an acoustically ambiguous sound, created by digitally mixing energy (via waveform averaging) from natural /ʃ/ and /s/ productions of each of the same 16 talkers. The **test** stimuli consisted of a 7-step continuum ranging from /aʃi/ to /asi/. Both the original ('typical') and the manipulated ('ambiguous') stimuli were annotated for two acoustic cues to fricative identity (the first and third peak in the amplitude over the frequency spectrum, P1 and P3; for more details, see Appendix A2).

2.2 Human categorization responses following exposure

40 native speakers of American English participated in each of the two conditions of the 16 experiments, for a total of 1,280 participants. Following exposure, each listener went through 10 cycles of 7-step test continua, with item presentation randomized within each cycle. This yielded a total of 89,600 post-exposure categorization responses (16 talkers * 2 bias conditions * 40 listeners * 7 steps along the test continuum * 10 cycles).

Appendix A3 reports how we analyzed the data from Cummings (2025). We found significant between-condition differences in categorizations of the same test items for 13 of 16 talkers, replicating the classic boundary shifts in LGPL. At the same time, the unprecedented scale also revealed considerable variability in the magnitude of effect sizes across experiments: how much listeners changed their categorization responses following exposure varied substantially between the 16 LGPL experiments.

Additionally reported in Cummings (2025) are a set of complementary experiments which include only the 2AFC test phase without biasing exposure (henceforth *test-only data* to distinguish it from the LGPL data). We use these independent test-only data to validate the categorization model described in Section 3.1.1.

3 Empirical indeterminacy, even at scale

We use the data from Cummings (2025) to test whether the exposure-elicited changes predicted by each of the three ASP mechanisms can be separated from each other, by comparing the fit of model predictions to human categorization responses.

3.1 Methods

3.1.1 Parameterizing the categorization model

Even before experimentally induced biases, listeners have expectations built-up over their lifetime for how talker-contingent (e.g. vocal tract length) and phonetic/co-articulatory factors affect fricative acoustics. To capture this, we used C-CuRE normalization (McMurray and Jongman, 2011) to remove talker's means from the acoustic cues to fricative identity (P1 and P3). Crucially, this step is distinct from the normalization change model evaluated below, which captures trial-by-trial adjustments to expected cue values over the course of the experiment.[1]

To estimate listeners' prior beliefs about category representations, we estimated $\mu_{c,0}$ and $\Sigma_{c,0}$ for $c \in \{/s/, /ʃ/\}$ over the normalized cues of the 40 clear, unmanipulated tokens of /s/ and /ʃ/ from each of the 16 talkers used in the LGPL studies (n = 1,280 pairs of P1-P3 values). Finally, we fit perceptual noise (Σ_ϵ) and lapse rates (λ) to the test-only data, as these parameters govern the categorization model independent of any exposure-induced changes. This reduces the number of parameters we had to fit to the LGPL data.

3.1.2 Parameterizing the change models

We assume that the parameterization of change models is identical across all 32 exposure conditions (16 talkers * 2 bias conditions). The parameters of the change models (κ_0 for normalization, $\kappa_{c,0}$ and $\nu_{c,0}$ for changes in category representations, or β_π for changes in decision-biases) were therefore fit to post-exposure categorization responses across all exposure conditions together.

We used five-fold cross-validated maximum likelihood to find the best-fitting parameters for each change model. Specifically, the data were folded between participants, so that the per-observation log-likelihoods we report below quantify the change model's ability to successfully generalize across participants. Instead of directly fitting change models against listeners' responses, we fit them against an estimate of listeners' response preferences *immediately following exposure* (for additional details, see Appendix A3). This decision was made because LGPL effects are known to diminish with repeated testing over uniform test continua (e.g., Liu and Jaeger, 2018; Cummings et al., 2026). This 'return to normal' is not something the change models we test are designed to capture.

Best-fitting parameters were similar across folds. Best fitting κ_0 across-folds averaged 66.3 (SD = 2.11). Best fitting $\kappa_{c,0}$ and $\nu_{c,0}$ averaged 48.9 (SD = 2.57) and 48.6 (SD = 1.48) respectively, and the best fitting β_π for the decision model averaged 0.12 (SD = 0.004).

3.2 Results

Figure 2 visualizes model fits to human behavior. Averaging across all 32 conditions, the three mechanisms perform similarly in capturing listeners' categorization responses following exposure.

To assess model distinguishability, we conducted pairwise t-tests comparing each pair of models' per-observation log-likelihoods across cross-validation folds, separately for each condition. Of 96 pairwise comparisons (32 conditions * 3 model pairs), only five reached significance after Bonferroni correction. Among these, no model was consistently favored over any other.

Figure 3 highlights four representative cases that span the range of outcomes. For M5./ʃ/, all

[1] We validated the decision to use normalized cues against the test-only data from Cummings (2025). Normalized cues significantly improved predictive accuracy for the test-only data, compared to unnormalized cues (Appendix A5).

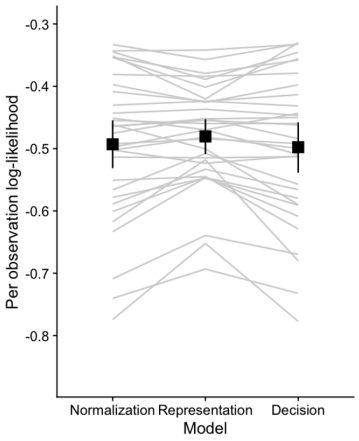


Figure 2: ASP change model fits across each of 32 exposure conditions, expressed in per-observation log-likelihood. Grey lines connect average fits per individual exposure condition over held-out post-exposure categorization responses across 5 folds of cross-validation. Points represent grand means over condition averages, with a 95% bootstrapped confidence interval.

three models provided statistically indistinguishable fits. 27 of 32 exposure conditions mirrored this trend. For exposure condition M6./ʃ/, the representation model systematically underpredicted the magnitude of the boundary shift, compared to the other two models. In contrast, changes in category representation provided the best numeric fits for exposure condition F5./s/ (though no differences reached significance following Bonferroni correction). Finally, for M2./ʃ/, changes in decision-biases provided a markedly worse fit than the other two models, which were themselves indistinguishable.

4 General Discussion

Recent findings suggest that some of the most frequently used experimental paradigms in research on speech perception provide comparatively little information about the adaptive mechanisms that they are meant to investigate (Xie et al., 2023). These findings were based on synthetic data from simulated experiments with simplified stimulus configurations. This leaves open whether the diagnosed empirical indeterminacy of existing paradigms is an artifact of the simplifying assumptions made by Xie and colleagues. The present work addressed this question directly, applying ASP to the largest existing LGPL dataset: 89,600 categorization responses from over 1,000 listeners across 32 exposure conditions. Our findings provide strong support for the conjecture provided

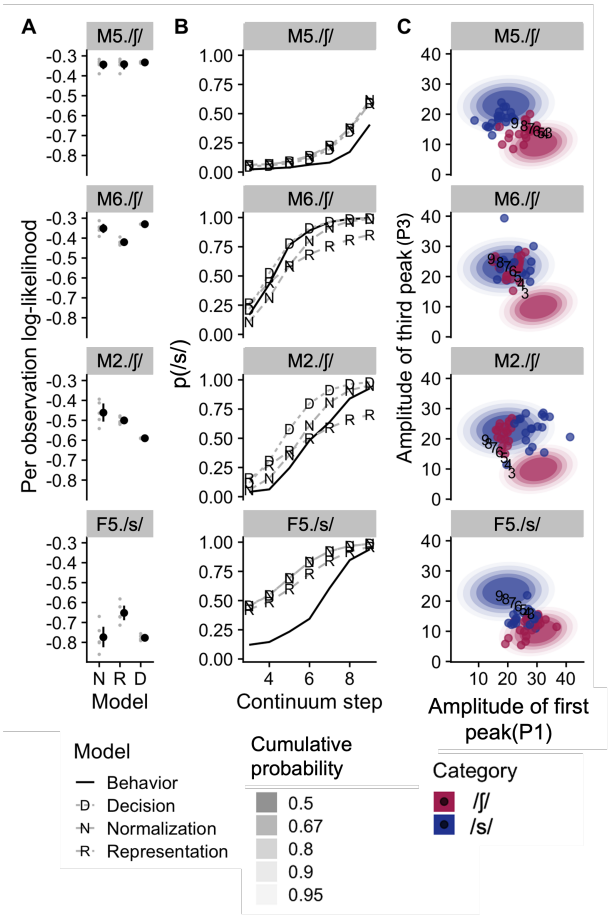


Figure 3: Divergent predictions of four representative exposure conditions (for all conditions, see Appendix A4 **Panel A:** Grey dots represent likelihoods within each fold of cross-validation; black pointranges show average likelihood and a 95% bootstrapped confidence interval over these folds. **Panel B**: Predicted perception of test continua by each model (averaged across 5 folds of cross validation, marked by the initial of the model employed) compared to human responses (solid line). **Panel C:** Input distributions by condition. Larger colored areas represent density distributions of listeners' assumed prior beliefs, colored by category. Points represent exposure input, colored by disambiguating lexical context. Black numbers represent the test tokens whose perception is shown in Panel B.

by Xie and colleagues: even for this large data set combining 16 different experiments, none of the three change models consistently outperformed the others. This should serve as a clear warning signal for future work: merely scaling up data collection within conventional experimental designs is unlikely to resolve the empirical indeterminacy between mechanisms within those paradigms.

At the same time, the scale and diversity of the Cummings (2025) data revealed important heterogeneity across exposure conditions. This provides

a glance at how much the degree of adaptivity observed in LGPL experiments might vary even if the exact same design, procedure, and approach to stimulus creation are used. For many of the exposure conditions, the predictions of the three change models were virtually indistinguishable; for some conditions, however, a subset of the three change models provided a superior fit. Importantly, when individual conditions did favor a particular mechanism, the winner was not always the representational change model—contrary to what is commonly assumed in the LGPL literature. In some cases, changes in representation yielded the worst predictions overall among the three change models (e.g., M6./ʃ/ in Figure 3). These patterns underscore a point that is easy to overlook when results are aggregated or when only a single talker is investigated (as in typical for the vast majority of LGPL experiments): the diagnosticity of an LGPL experiment depends heavily on the acoustic properties of the specific stimuli used—properties that are typically not controlled or even reported.

4.1 Tracing the indeterminacy to stimulus design

If elucidating the mechanisms of adaptive speech perception is the goal, how might one proceed? One possibility, raised in the introduction, is that 2AFC tasks are simply not sensitive enough to detect subtle behavioral differences caused by distinct adaptation mechanisms. If this is true, the field will face a formidable challenge: the 2AFC is not merely a convenience but a foundation on which theories across speech perception, phonetic acquisition, category learning have been built. Abandoning it would sever connections to decades of cumulative evidence across these subfields.

The divergence maps presented in Figure 4 point to a more optimistic alternative. To create these maps, we computed the pairwise difference between change models predictions over the acoustic space—the predicted probability of an /s/-response immediately after exposure, depending on the acoustic cues (P1 and P3). This process was repeated for all pairs of change models and for each of the 16 talkers. This reveals where (i.e., in which part(s) of the acoustic space) the three change models make distinct predictions.

The divergence maps of the two exposure conditions in Figure 4 offer two important insights. First, for both exposure conditions, diagnosticity could be improved by selecting different test stimuli. The test continua used in Cummings (2025) (open circles) occupy regions of the acoustic space where power to detect a between-condition difference is high (indicated by large tile size), but where the three change models make similar predictions (indicated by light coloring). In other words, the

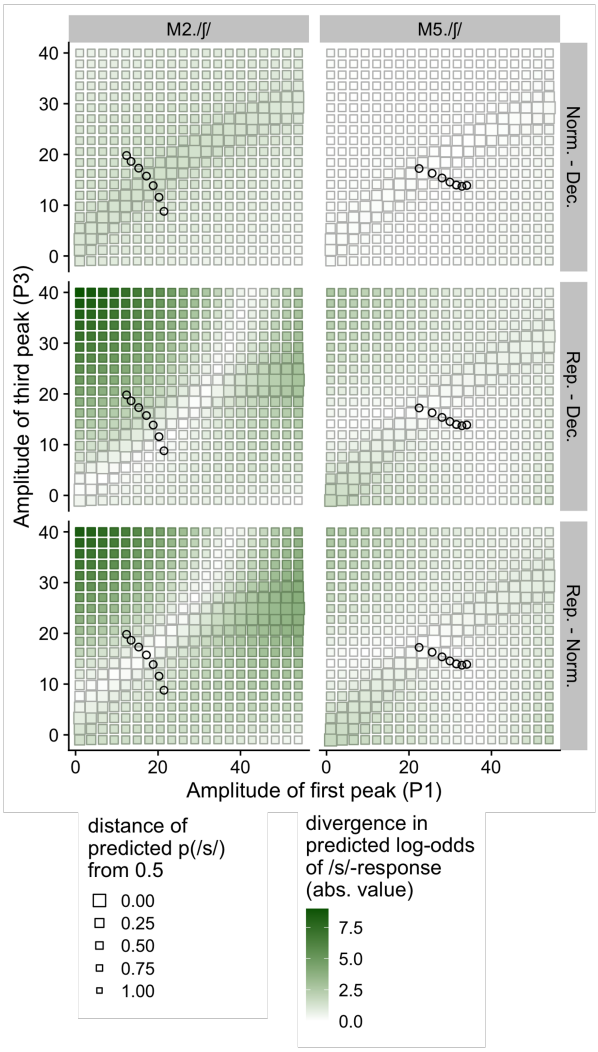


Figure 4: Predicted diagnosticity of different stimuli in the Cummings (2025) data. Shown is the predicted divergence in change model predictions across the acoustic space after two different exposure conditions (M2./ʃ/ and M5./ʃ/, columns). Each model was simulated 100x, with parameter values drawn from a normal distribution centered on the best fitting parameters to the human LGPL data and with variance drawn from the average estimated parameter uncertainty across the cross-validation folds. Diagnosticity is higher for more intense colors (larger differences between model predictions) and larger tiles (power to *detect* the difference in predictions is maximized at $E[p(/s/\text{-}response)]$). Open circles show the continuum steps presented in Cummings (2025). Acoustics for the two exposure condition are shown in Figure 3C.

conventional test tokens are well-placed for detecting *that* adaptation occurred, but poorly placed for determining *which* mechanism underlies it. A notable exception to this are the test tokens for M2./ʃ/, which cross the space that has the highest diagnosticity for the comparison of normalization vs. changes in decision-making (top left panel). Even for this exposure condition, however, diagnosticity for the other two pairwise comparisons would have been improved by selecting test stimuli elsewhere in the acoustic space (at regions with darker colors). For example, normalization and representational changes, which were indistinguishable in Figure 3A, could become separable if the experimenter were to use test tokens placed in the high-divergence regions (e.g., the large green squares in the bottom left panel of Figure 4). For M5./ʃ/, the same logic applies, though the expected gains are more modest.

Second, the two exposure conditions differ markedly in how much diagnosticity any test token placement can achieve. Across the entire acoustic space, M2./ʃ/ yields larger divergences between models than M5./ʃ/. This means that the choice of exposure input—not just test tokens—also constrains diagnosticity.

Taken together, these observations suggest that the empirical indeterminacy documented above reflects the stimulus design, rather than being a principled limitation of 2AFC tasks. The models' predictions converge not because the behavioral measure lacks resolution, but because competing mechanisms happen to make similar predictions for the particular acoustic tokens used during the exposure and test phases of the experiment. These observations point towards two concrete changes in experimental practice.

4.2 Actionable recommendations

First, **selection of test token should be guided by model-predicted divergence rather than perceptual ambiguity alone**. The standard practice for LGPL experiments of constructing a continuum between two category endpoints aims at ensuring that experimenters can detect the *presence* of adaptive changes (differences between the two exposure conditions of an LGPL experiment). However, this does not guarantee that the experiment will be diagnostic of the *mechanisms* underlying those changes. We propose that test tokens be selected according to three criteria: (i) high predicted divergence between models, (ii) avoidance of floor or ceiling response rates (where power to detect effects in 2AFC is lowest), and (iii) sufficient acoustic spread to mitigate concerns about perceptual anchoring or repetition fatigue. This change requires no modification to the task, the exposure phase, or the sample size.

Second, echoing a recommendation made in Xie et al. (2023), **researchers should examine and report the acoustic properties of their stimuli, and use model-guided simulation to evaluate diagnosticity before data collection**. It is not standard in LGPL or accent adaptation research to report acoustic cue values for exposure or test stimuli. Yet our results demonstrate that conditions vary drastically in their ability to distinguish between the mechanisms, even when identical methods are used. As illustrated by the M5./ʃ/ condition above, some exposure inputs inherently limit diagnosticity regardless of test token placement. Researchers cannot know which situation they are in without examining the acoustics and computing divergence maps, or at a minimum, simulating their experiment within ASP or a comparable computational framework. We submit that this kind of pre-experimental diagnosticity check should become routine.

Throughout this paper, we have exclusively compared models wherein adaptation is attributable to a single mechanism. Having established that the current data are not reliably diagnostic even between these extreme scenarios, they certainly cannot identify where a listener falls within the larger space of possible mechanism mixtures (Figure 1C). Nevertheless, the tools developed here are prerequisites for addressing identification weighted mixtures of mechanisms. Determining that each mechanism makes divergent predictions in isolation is a necessary first step before assessing what combination might best account for a given phenomenon.

In conclusion, the 89,600 categorization responses analyzed here carry strikingly little information about which adaptive mechanisms underlie the observed behavior, despite the fact that these mechanisms differ starkly in the computations they involve. The divergence maps provided in Figure 4 suggest that this indeterminacy is not a limitation of the behavioral task itself. It is, at least in part, a consequence of stimulus selection practices that can be improved with relatively modest, model-guided changes. We hope that the framework and tools presented here provide a foundation for this next phase of research—one focused not on whether

adaptation occurs, but on how.

Acknowledgments

Research reported in this work was supported by NIH-NICHD grant R01HD111936.

References

Keith S. Apfelbaum, Ethan Kutlu, Bob McMurray, and Efthymia C. Kapnoula. 2022. Don't force it!: Gradient speech categorization calls for continuous categorization tasks. *The Journal of the Acoustical Society of America*, 152:3728–3745.

Santiago Barreda and T. Florian Jaeger. 2025. Reintroducing and testing the probabilistic sliding template model of vowel perception. *Linguistics Vanguard*.

Tessa Bent and Melissa M Baese-Berk. 2021. Perceptual learning of accented speech. *The Handbook of Speech Perception*, pages 428–464.

Paul-Christian Bürkner. 2017. Advanced bayesian multilevel modeling with the R package brms. *Preprint*, arXiv:1705.11123.

Shawn Cummings. 2025. *Linking lexically guided perceptual learning to statistical patterns in speech input*. Ph.D. thesis, University of Connecticut.

Shawn N. Cummings, Emma C. Hodges, and Rachel M. Theodore. 2026. Cumulative input sensitivity predicts both attenuation and stability of lexically guided perceptual learning. *Psychonomic Bulletin & Review*, 33:116.

Olivia Guest and Andrea E. Martin. 2021. How computational modeling can force theory building in psychological science. *Perspectives on Psychological Science*, 16:789–802.

Jingyuan Huang and Lori L. Holt. 2009. General perceptual contributions to lexical tone normalization. *The Journal of the Acoustical Society of America*, 125:3983–3994.

Alexandra Jesse. 2021. Sentence context guides phonetic retuning to speaker idiosyncrasies. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 47:184–194.

Keith Johnson and Matthias J Sjerps. 2021. *Speaker normalization in speech perception*, chapter 6. John Wiley & Sons, Ltd.

Allard Jongman, Ratree Wayland, and Serena Wong. 2000. Acoustic characteristics of english fricatives. *The Journal of the Acoustical Society of America*, 108(3):1252–1263.

Tanya Kraljic and Arthur G Samuel. 2005. Perceptual learning for speech: Is there a return to normal? *Cognitive Psychology*, 51:141–178.

Erika JC Laing, Ran Liu, Andrew J Lotto, and Lori L Holt. 2012. Tuned with a tune: Talker normalization via general auditory processes. *Frontiers in psychology*, 3:203.

Linda Liu and T Florian Jaeger. 2018. Inferring causes during speech perception. *Cognition*, 174:55–70.

Bob McMurray and Allard Jongman. 2011. What information is necessary for speech categorization?: Harnessing variability in the speech signal by integrating cues computed relative to expectations. *Psychological Review*, 118:219–246.

Emily B Myers and Laura M Mesite. 2014. Neural systems underlying perceptual adjustment to non-standard speech tokens. *Journal of Memory and Language*, 76:80–93.

Terrance M. Nearey and Peter F. Assmann. 2007. Probabilistic 'sliding-template' models for indirect vowel normalization. In Maria-Josep Solé, Patrice Speeter Beddor, and Manjari Ohala, editors, *Experimental Approaches to Phonology*, pages 246–269. Oxford University Press, Oxford.

Allen Newell. 1973. You can't play 20 questions with nature and win: Projective comments on the papers of this symposium. In William G. Chase, editor, *Visual Information Processing*, pages 283–308. Academic Press, New York.

Dennis Norris, James M McQueen, and Anne Cutler. 2003. Perceptual learning in speech. *Cognitive Psychology*, 47:204–238.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Odette Scharenborg and Esther Janse. 2013. Comparing lexically guided perceptual learning in younger and older listeners. *Attention, Perception, & Psychophysics*, 75(3):525–536.

Maryann Tan and T. Florian Jaeger. 2025. Learning to understand an unfamiliar talker: Testing distributional learning as a model of rapid adaptive speech perception. *Cognition*, 265:106195.

Kodi Weatherholtz and T Florian Jaeger. 2016. Speech perception and generalization across talkers and accents. *Oxford Research Encyclopedia of Linguistics*.

Xin Xie, T. Florian Jaeger, and Chigusa Kurumada. 2023. What we do (not) know about the mechanisms underlying adaptive speech perception: A computational review. *Cortex*, 166:377–424.

Tal Yarkoni and Jacob Westfall. 2017. Choosing prediction ver explanation in psychology: Lessons from machine learning. *Perspectives on Psychological Science*, 12:1100–1122.

A Appendix

A1: Lexically guided perceptual learning and implementation under ASP

See Figure 5, below.

A2: Acoustic measurements

Many acoustic cues have been posited to underlie the distinction between /s/ and /ʃ/, with spectral center of gravity being a popular choice (e.g., Jongman et al. 2000). /s/ is characterized by high energy at high frequencies (above 6.5 kHz), while /ʃ/ tends to have more energy concentrated below 3.5 kHz (aligned with vowel formants). To characterize our tokens, we extracted the amplitude of the spectral peak in each of these ranges for each of our tokens. These cues necessarily correlate highly with spectral center of gravity (as tokens with more energy at a higher spectral peak will have a higher spectral average). See Figure 6, below.

All exposure and test tokens were annotated based on these cues, which were used as input to the ASP model (shown in Figure 7, below).

A3: Findings of Cummings (2025)

Figure 8 summarizes the results of Cummings (2025), separately for each of the 16 talkers. The LGPL effects shown in this figure were obtained from a Bayesian mixed-effects logistic regression to estimate listeners' responses *immediately following exposure*. We used package `brms` (Bürkner, 2017) in R (R Core Team, 2021) to predict the probability of /s/-responses from as a function of continuum step and test block, as well as random by-participants intercept and slopes for all effects. We treated both continuum step and test block as monotonic, potentially non-linear, effects using function `mo()` of the `brms` package. The effects we report are those estimated for the *0th* test trial, i.e., participants' behavior before the first test trial.

For the fitting of change models, we used these estimates of $p(/s/\text{-}response|continuum\ step, test\ block, participant)$ before any effects of repeated testing. Specifically, we obtained one estimate per listener and continuum step (instead of the 10 observations per listener and continuum step). This approach has the additional upside that it avoids inflating the evidence available to change models since the 10 responses per listener and continuum steps are *not* independent observations. It should be noted, however, that our approach is conservative and likely somewhat under-estimates the available amount of evidence. An alternative, better, approach would be to develop hierarchical implementations of change models that can adequately handle repeated-measures data from participants, while also extending the models to account for the effects of repeated testing. This is a non-trivial effort to be pursued in future work.

A4: Individual conditions as simulated by ASP

Extensions of Figure 3 to include all 32 input conditions are shown in Figures 9 and 10, below.

A5: Normalized and unnormalized token distributions

See Figure 11, below.

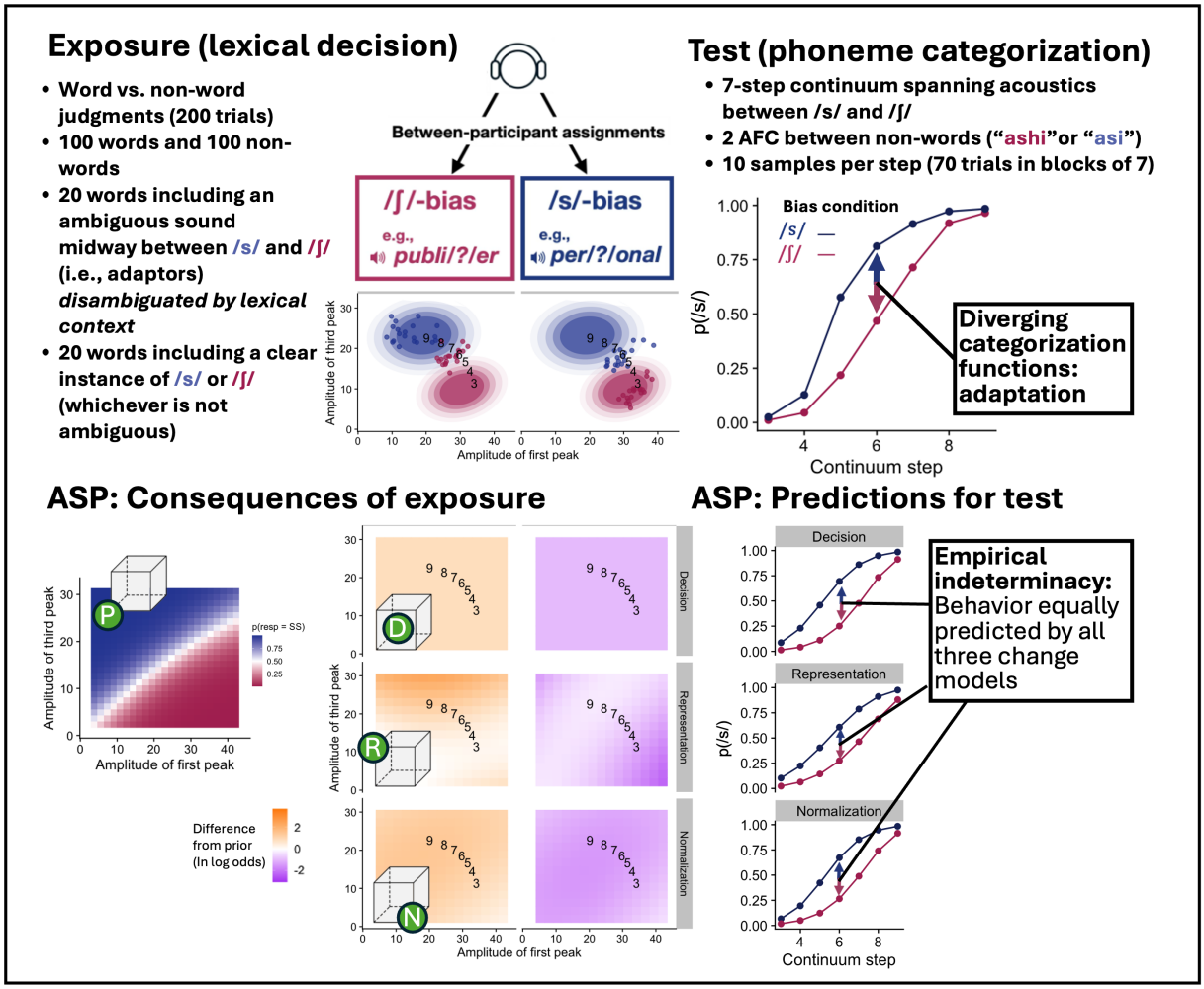


Figure 5: Schematic of the lexically guided perceptual learning (LGPL) paradigm, and simulation under ASP. In the scatter plot under 'Exposure', colored dots represent the acoustics of fricatives in exposure, while black numbers designate the acoustics of the continuum presented in test. Filled areas show listeners' assumed long-term expectations about cue-category mappings (see subsection 3.1). The leftmost panel under 'ASP' shows predictions of the categorization model, and the right panels show differences in expected categorizations after integrating exposure input (assuming only a single change mechanism is active).

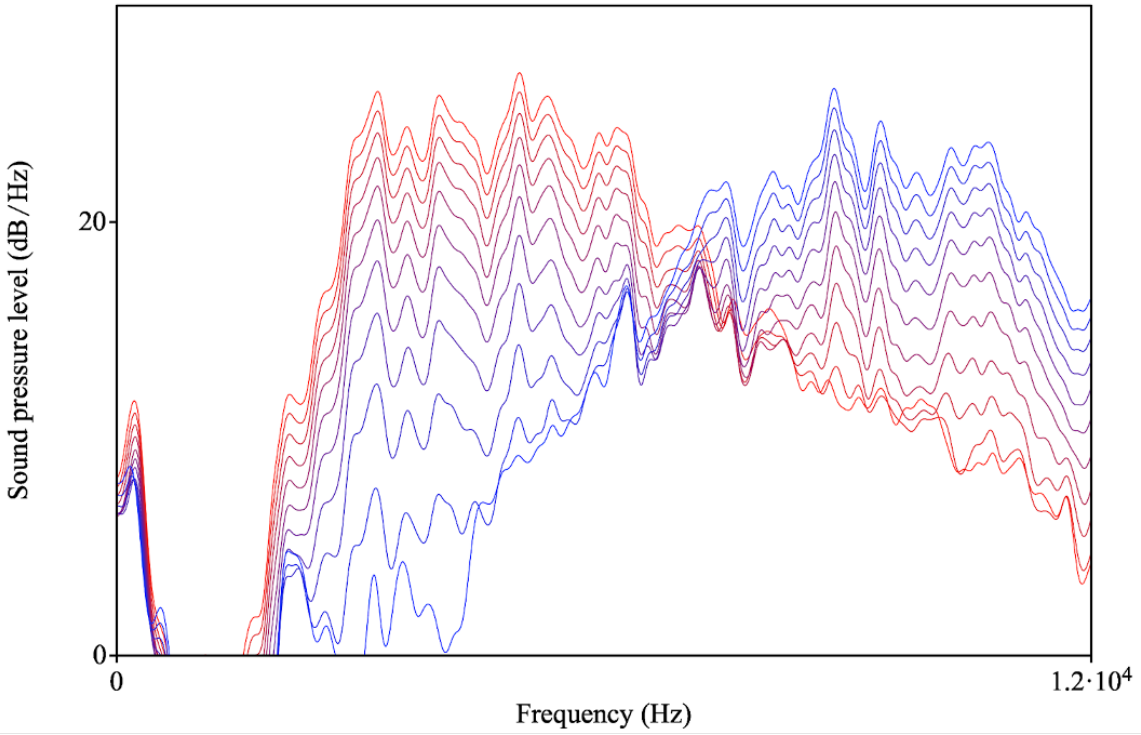


Figure 6: Long-term average spectra for 11 steps spanning a continuum from /s/ to /ʃ/ (as recorded in “asi” and “ashi” by Talker F1). Blue and red endpoints represent clear /s/ and /ʃ/ segments. As tokens become more /ʃ/-like, energy in the 1.9–4.5 kHz range diminishes and energy in the 8.5–11 kHz range increases.

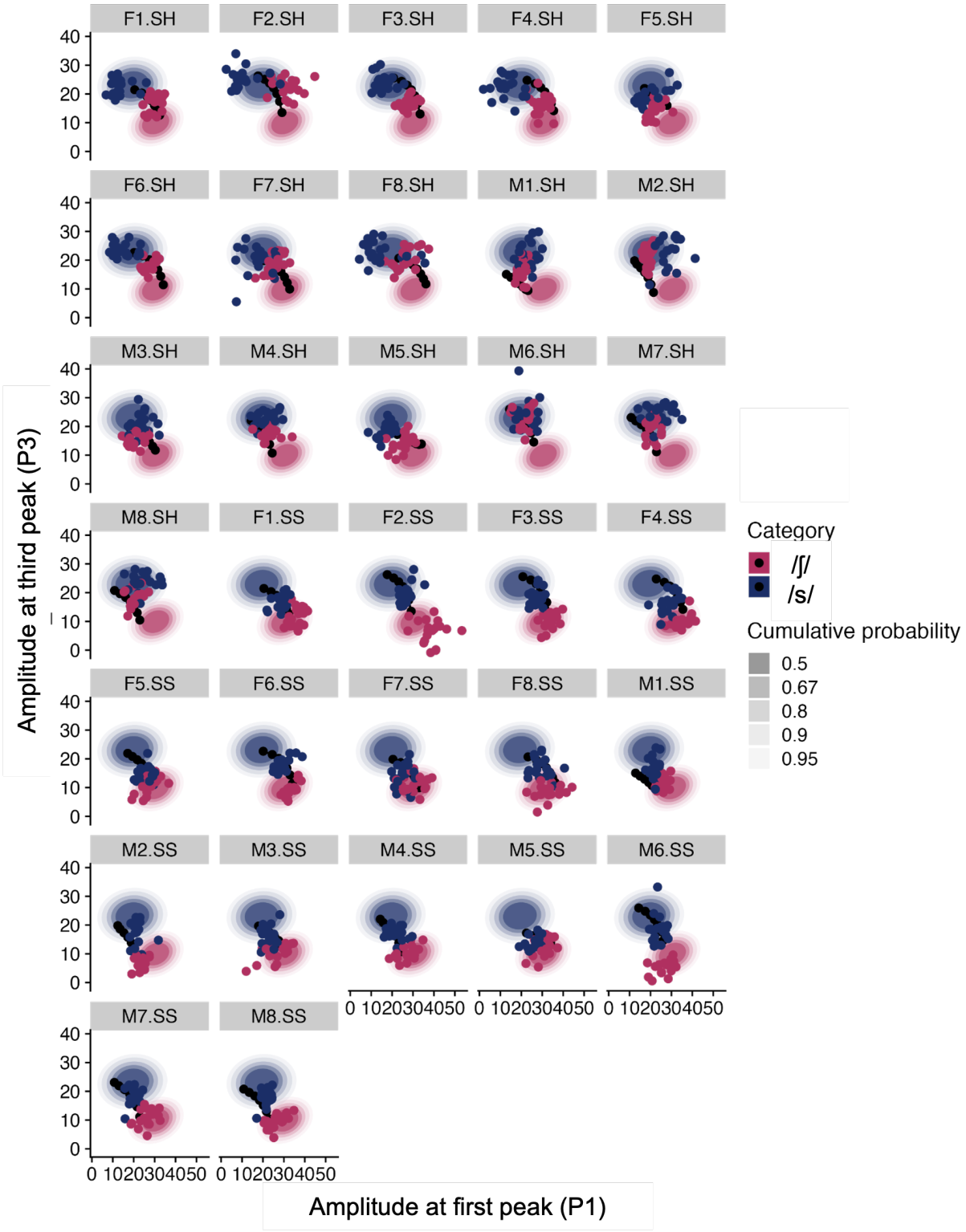


Figure 7: Extension of Panel A of Figure 2 in main text, showing all conditions. Input distributions by condition. Larger colored areas represent density distributions of listeners' assumed prior beliefs, colored by category. Colored dots represent exposure input, colored by disambiguating lexical context. Black points represent tokens presented in test.

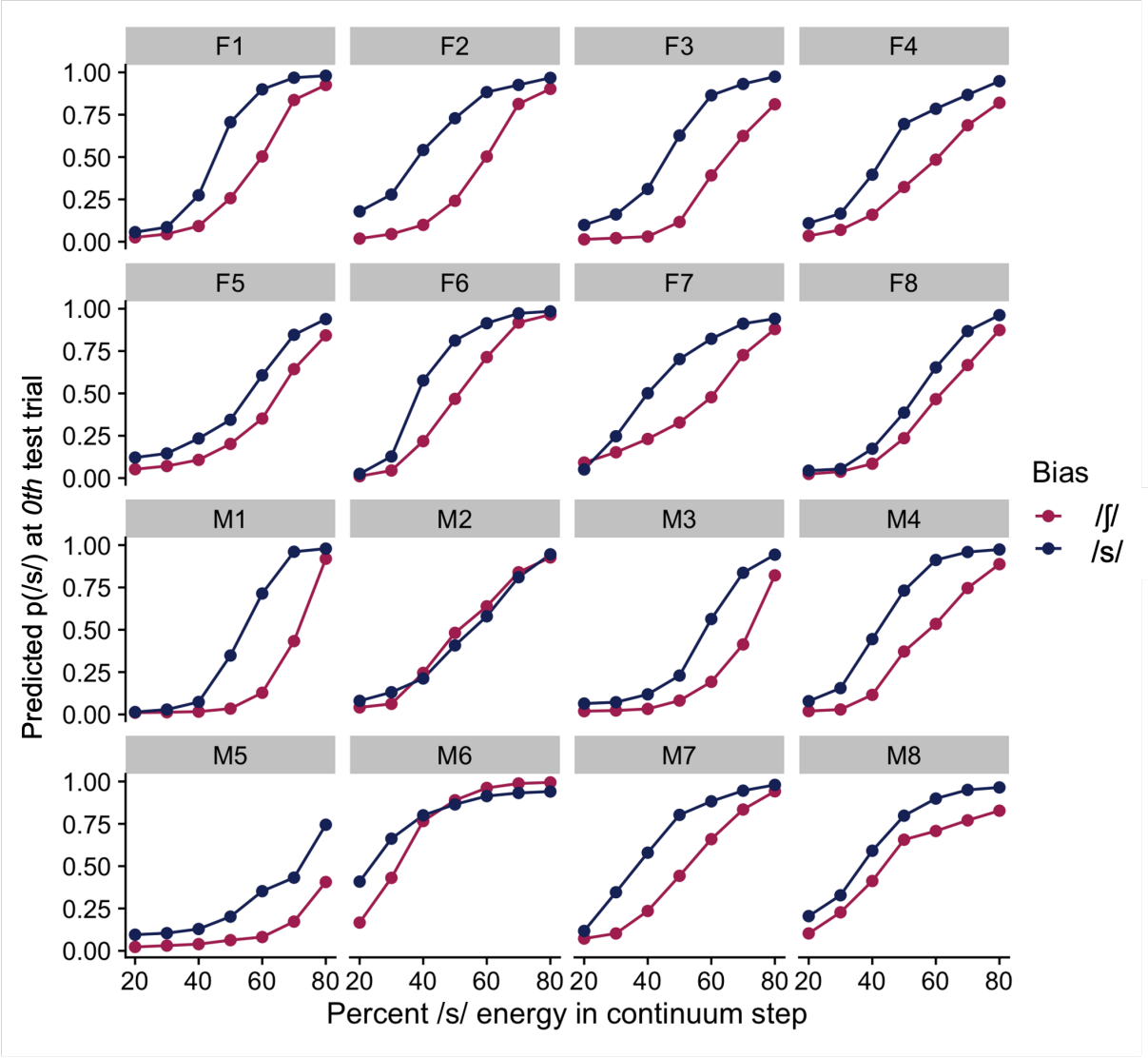


Figure 8: Results across conditions from Cummings (2025), showing an estimate of listeners' categorization responses before repeated testing could affect them.

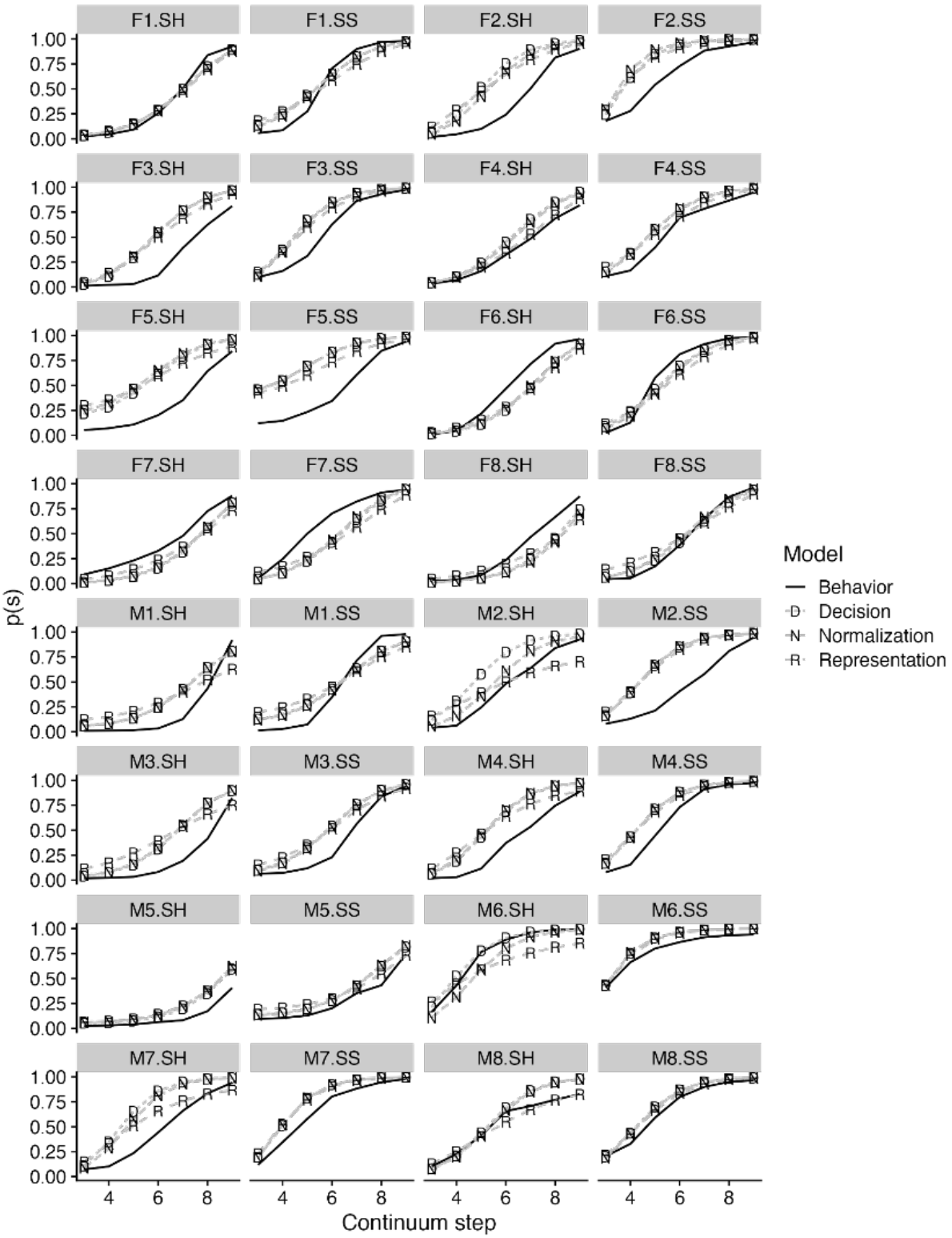


Figure 9: Extension of Panel B of Figure 3 in main text, showing all conditions. Predicted perception of test continua by each model (averaged across 5 folds of cross validation, marked by the initial of the model employed) compared to human responses (solid line).

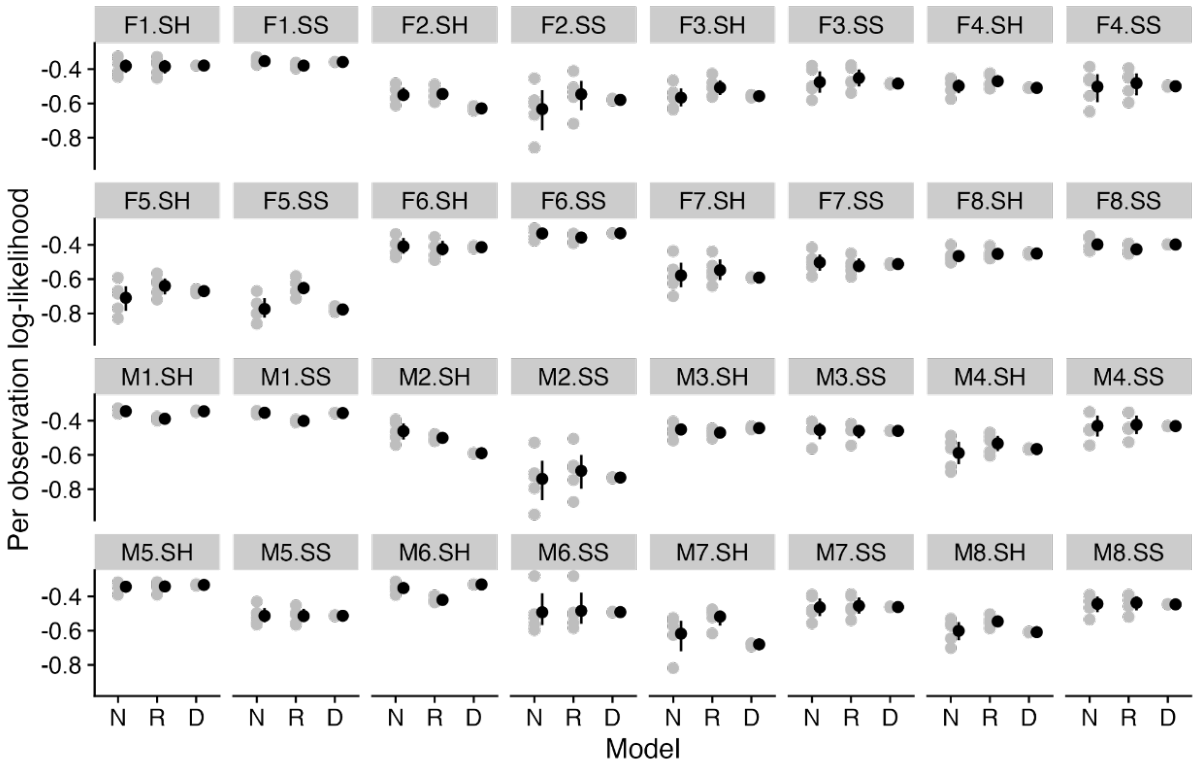


Figure 10: Extension of Panel A of Figure 3 in main text, showing all conditions. Grey dots represent likelihoods within each cross-validation fold; black pointranges show average likelihood and error as a 95% bootstrapped confidence interval.

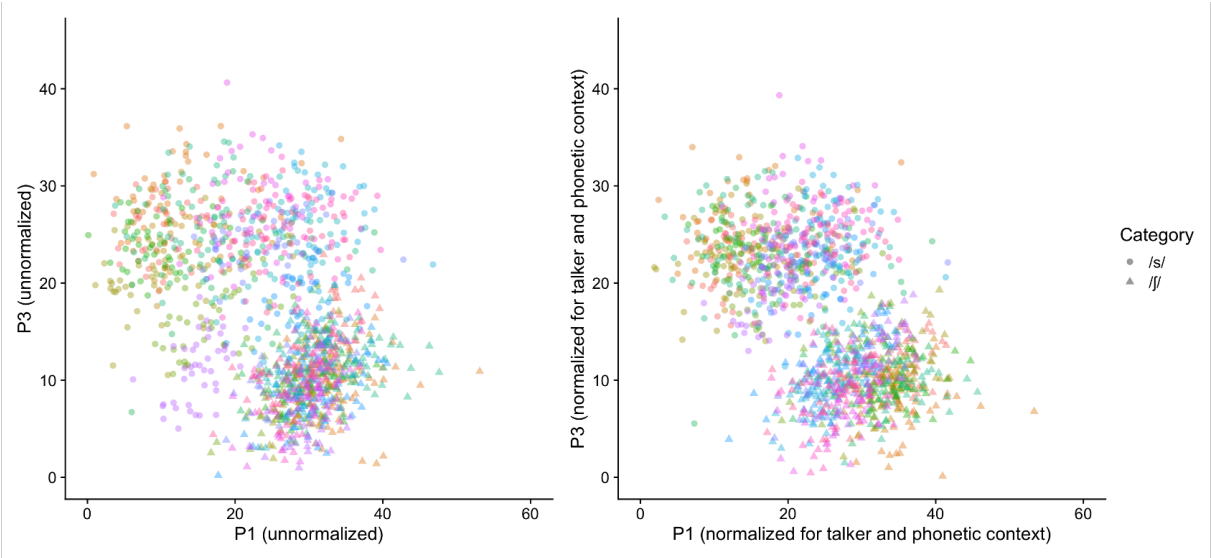


Figure 11: Normalized (right) and unnormalized (left) token distributions. Triangular tokens designate /s/, while circles designate /ʃ/. Color indicates talker. All simluations reported in main text were based on normalized acoustics.

Modeling generalization in perceptual learning of speech

Yiming Lu
Basque Center on Cognition, Brain and Language
y.lu@bcbl.eu

Xinyu Leslie Liao
University of Toronto
xinyuleslie.liao@mail.utoronto.ca

Alejandro Tabas
Basque Center on Cognition, Brain and Language
a.tabas@bcbl.eu

Xin Xie
University of California, Irvine
xxie14@uci.edu

Abstract

A hallmark of learning is generalization to novel instances. In speech, exposure to atypical pronunciation drives perceptual adjustment that can generalize to unheard tokens. Prior work has attributed constraints on generalization primarily to acoustic similarity between exposure and test contexts. We propose that generalization can also be understood as an inference problem: listeners must determine whether, and how strongly, a learned phonetic mapping should apply in a new context. We test this proposal using data from a recent experiment in which listeners were exposed to shifted vowel pronunciations and then tested on minimal pairs varying in lexical frequency. Learning effects appeared strongest when the exposure direction aligned with a high-frequency alternative in mixed-frequency pairs, and were absent for low-frequency pairs. The observed pattern could reflect token-level acoustic similarity, reliance on prior expectations, or frequency-dependent constraints in applying the learned mapping. We formalized these alternatives within a Bayesian belief-updating framework: a talker-specific model assuming full transfer, a mixture-of-expectations model that interpolates between the updated representation and the listener's prior, and a hierarchical Bayesian model that deploys the updated representation with uncertainty. The talker-specific model captured most generalization patterns through its sensitivity to token-level acoustic properties, but overpredicted learning for low-frequency pairs. The hierarchical model best recovered the theoretically central exposure-control contrast pattern, suggesting that lexical frequency may constrain how learned representations are applied. Our results provide a computationally explicit framework for studying how contextual factors shape generalization in speech perception.

Keywords: Perceptual learning, computational modeling, generalization, lexical frequency, acoustic similarity

1 Introduction

Listeners rapidly adjust their speech perception in response to novel pronunciation patterns, demonstrating remarkable adaptivity (Norris et al., 2003; Idemaru and Holt, 2011; Clarke and Garrett, 2004; Bradlow and Bent, 2008; Maye et al., 2008). One prominent example of this adaptivity is manifested via lexically-guided perceptual learning (Norris et al., 2003): brief exposure to an ambiguous sound in a lexically biasing context—such as a vowel midway between /ɪ/ and /ɛ/ embedded in the word "l[ɛ]mon", where /ɛ/ is the only viable interpretation—shifts how listeners subsequently categorize tokens along the relevant /ɪ/-/ɛ/continuum such that more tokens are perceived as /ɛ/ following exposure. This learning effect is robust across phonemic contrasts and languages (Kraljic and Samuel, 2006; Chládková et al., 2017; Mitterer et al., 2011).

A central question is how such learning generalizes beyond the immediate exposure context. Prior work has extensively investigated generalization across talkers (Bradlow and Bent, 2008; Alexander and Nygaard, 2019; Baese-Berk et al., 2013; Xie and Myers, 2017; Xie et al., 2021; Aoki and Zellou, 2025a,b), phonetic contexts and word positions (Jesse and McQueen, 2011; Mitterer and Reinisch, 2017; Bowers et al., 2016), asking whether adjustments learned during exposure transfer to test. In this literature, similarity between exposure and test has been hypothesized to be a primary explanatory construct: generalization succeeds when the test context is sufficiently similar to exposure, and fails otherwise (Kraljic and Samuel, 2006; Eisner and Mcqueen, 2005; Reinisch and Holt, 2014; Xie and Myers, 2017; Xie et al., 2021). However, similarity has been conceptualized at different levels. At the broadest level, generalization has been shown to depend on global talker similarity — whether the test voice matches the exposure voice (Eisner et al.,

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 529–541
July 3-4, 2026 ©2026 Association for Computational Linguistics

2013; Reinisch and Holt, 2014). At a finer grain, listeners may evaluate the acoustic similarity between specific test tokens and the distributions encountered during exposure, such that tokens closer in the acoustic space to the exposure input are more readily accepted into the updated categories (Chládková et al., 2017; Jin et al., 2025). Yet similarity defined at these levels do not always capture the empirical pattern (Lai and Tamminga, 2024), and existing accounts also leave open the question of how similar is similar enough to drive generalization. In the current paper, we investigate the possibility that beyond acoustic similarity between exposure and test tokens, generalization is further constrained by a listener's confidence in deploying the learned mapping in a given test context, which depends on additional contextual factors.

A recent experiment (Liao and Kang, 2026) (henceforth LK26) provides a particularly informative case. In LK26, listeners were exposed to an ambiguous vowel between /ɪ/ and /ɛ/ embedded in high-frequency, non-minimal-pair words (e.g., *kitchen*), with the ambiguity biased toward either /ɪ/ (lowering group) or /ɛ/ (raising group), along with a control group hearing canonical pronunciations of both vowels. At test, participants categorized tokens from a five-step vowel continuum embedded in /ɪ/-/ɛ/ minimal pairs. Critically, the test pairs varied in the lexical frequency of the /ɪ/-consistent and /ɛ/-consistent words, yielding four frequency configurations: high-high (HH; e.g., *sit-set*), high-low (HL; e.g., *kick-keck*), low-high (LH; e.g., *din-den*), and low-low (LL; e.g., *trimmer-tremor*). The results revealed that exposure-induced categorization shifts were not uniformly expressed across test conditions (Figure 1). In the mixed-frequency conditions (HL and LH), shifts relative to the control condition were observed only when the direction of exposure bias aligned with the high-frequency member of the pair: in HL, the lowering group differed from control but the raising group did not; in LH, the reverse held. In the LL condition, neither biasing group differed reliably from control, indicating an absence of detectable learning effects when both lexical alternatives were low-frequency. In HH, the lowering effect was more clearly supported than the raising effect, despite identical exposure structure.

These findings raise an important question: do the differences across frequency conditions reflect differences in the learning process itself, or differences in how a learned phonetic mapping is applied

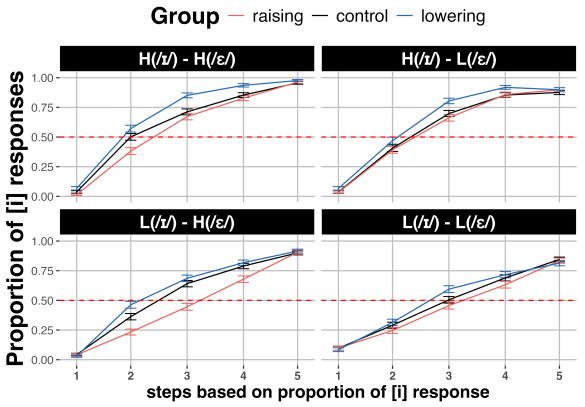


Figure 1: Results of the LK26 experiment. Points indicate the mean proportion of /ɪ/-response. Error bars indicate the standard error of the mean across subjects. On the x-axis, step 1 represents a more /ɛ/-like stimulus, and step 5 represents a more /ɪ/-like stimulus. The distributions of exposure and test items are shown in Appendix Figure 3 and 4.

at test? Several observations narrow the space of explanations. First, the same talker produced all exposure and test items, so differences across frequency conditions cannot be attributed to talker identity. Second, a frequency-based similarity account—predicting generalization only when the test frequency bias matches exposure—captures the HL, LH, and LL patterns but cannot explain why the lowering group generalized in HH, where neither alternative had a frequency advantage. Third, if lexical frequency operated as a general response bias favoring high-frequency words, exposure effects should appear regardless of whether the high-frequency member aligns with the direction of the shift. Yet the data show that only the shift aligned with the high-frequency alternative was expressed. Together, these observations suggest that lexical frequency did not operate as a simple similarity dimension or a global response bias.

Because the exposure items were held constant for each group, it is reasonable to believe that exposure-elicited learning did not vary across frequency conditions. What varied was how learning was expressed as a function of word frequency. This points to a mechanism that modulates the *expression* of learning, rather than learning itself.

In this paper, we ask whether, beyond acoustic similarity, lexical frequency may serve as the contextual factor that modulates the expression of learning. A working hypothesis is that high-frequency words, supported by more robust mental representations (Pierrehumbert, 2016; Connine

et al., 1993; Todd et al., 2019), may provide a sufficiently stable lexical context for the listener to confidently deploy a recently learned phonetic adjustment; low-frequency words, with less well-specified representations, may not. We propose that generalization of perceptual learning can be understood as an inference about the *applicability* of a learned phonetic mapping to a novel test context. In what follows, we formalize the process of *generalization under uncertainty* in three computationally distinct models and evaluate the three models against the LK26 data. Our goal is to determine what kind of mechanism gives rise to the observed empirical patterns—specifically, the selective expression of learning as a function of the frequency structure of test items. In doing so, we seek to contribute a computationally explicit framework to the broader literature on how learned phonetic adjustments generalize beyond the contexts in which they were acquired.

2 Possible strategies of generalization

How should we conceptualize the problem that listeners are trying to solve? We contend here that, similar to perceptual learning, generalization behavior can be understood in a general framework of perceptual inferences under uncertainty. Upon exposure to novel pronunciation, listeners update their internal generative models. During generalization, listeners must infer which generative model is most relevant to the current context, as well as the degree to which that model should guide perception (Xie and Myers, 2017; Xie et al., 2021; Tan and Jaeger, 2025).

One possibility is that listeners treat generalization trials as a re-encounter with a previously experienced situation, directly applying the updated model to novel tokens. Evidence for such full transfer comes from studies showing that learned boundary shifts generalize across word positions without attenuation (Jesse and McQueen, 2011). This scenario can be modeled through direct application of a Bayesian ideal adaptor updated during exposure (Kleinschmidt and Jaeger, 2015), widely employed to account for adaptation across various paradigms, including selective adaptation (Kleinschmidt and Jaeger, 2015), accent adaptation (Tan et al., 2021), cue reweighting (Lu and Xie, 2024), and lexically-guided perceptual learning (see a comprehensive review in Xie et al. (2023)).

However, generalization is not always complete. Listeners sometimes show little transfer despite successful learning—for instance, when the learned pattern is produced by a speaker for whom it is unexpected (Eisner et al., 2013). Two computational mechanisms could give rise to such constrained generalization. First, listeners may have strong prior expectations that compete with the updated model, effectively falling back on longer-term experience. Kleinschmidt and Jaeger (2015) allude to this mechanism: categorization can be determined by combining predictions of multiple generative models weighted by their inferred relevance, so that limited generalization arises from a stronger influence of the prior. We formalize this as a mixture-of-expectations model, in which responses reflect a weighted combination of the exposure-updated representation and the listener's prior knowledge.

Second, listeners may still adopt the updated model but deploy it with greater uncertainty. This intuition can be formalized within a hierarchical Bayesian framework, which has been widely used to explain phenomena such as human concept learning (Kemp et al., 2007), linguistic coordination and convention (Hawkins et al., 2023) and syntactic priming (Xu and Futrell, 2024). Related ideas have also been proposed in speech perception, although they have either not been directly implemented (Pajak et al., 2016; Kleinschmidt and Jaeger, 2015), or have been implemented in contexts that differ substantially from the present case (Pajak et al., 2013; Hosier and Bicknell, 2018; Nielsen and Wilson, 2008). Following Xu and Futrell (2024), we model listeners as first updating a higher-level talker-specific representation through exposure, then generating a lower-level model for novel test contexts by drawing on this higher-level representation. A hyperparameter controls how tightly the lower-level model draws on the updated representation: larger values yield stronger pooling (more confident deployment), while smaller values permit greater deviation (less confident deployment). In this way, limited generalization arises from higher uncertainty about the applicability of the updated model, rather than from reliance on a competing prior.

3 Modeling

We now provide the mathematical formulation for the three candidate mechanisms introduced above. All three share a common learning component—a Bayesian belief-updating model that estimates

updated phonetic category representations from exposure—and differ only in how the learned representation is expressed at test. In other words, the models implement alternative assumptions about how listeners decide whether the exposure-updated representation should be applied when categorizing test tokens in a new lexical context. In both constrained models (mixture-of-expectations and hierarchical Bayesian), the transfer parameter is allowed to vary by frequency, capturing the hypothesis that high-frequency test contexts support more confident deployment than low-frequency ones. Next, we describe the steps taken to obtain model prediction, beginning with the estimation of listeners' prior knowledge, before turning to data pre-processing, confound control, model details, and optimization procedures.

3.1 Estimating prior knowledge

We adopted the assumption that our participants' baseline knowledge reflects cumulative experience with speech in their native environment. To approximate this experience, we pooled together three Canadian English corpora, reported in a cross-dialect analysis of English /ɪ/–/ɛ/ allophony (Smith et al., 2025). We removed outliers on both the F1 and F2 dimensions, excluding tokens more than 3 standard deviations from the mean. Within each talker, we sampled an equal number of tokens from each category to ensure balanced representation. We excluded talkers who contributed fewer than the median of the distribution (104 tokens), and randomly sampled 104+ tokens from each category for each of the 96 remaining talkers, resulting in a total of 10,008 observations. The resulting corpus is shown in Appendix Fig. 5.

3.2 Pre-processing

We converted formant frequencies to the Mel scale and applied a talker-normalization scheme (McMurray and Jongman, 2011). F1 and F2 were centered relative to each talker's mean cue value. We performed this procedure on all the tokens, including the experiment tokens and those from the corpus used to estimate prior knowledge.

3.3 Category prior and version bias

The test continuum was previously calibrated so that the midpoint corresponded to equal /ɪ/ and /ɛ/ response rates, but listeners in this sample showed more /ɪ/ responses than expected in three of four conditions (HH, HL, LH in Fig. 1). This could reflect an asymmetric category prior or a bias from residual acoustic properties of the source words used to synthesize the test tokens. To estimate these confounds, we fit a generalized linear mixed-effects model to the first 20 test trials from each control-group participant, restricting the window to minimize influence of any additional learning during the test phase (Tan and Jaeger, 2025). Fixed effects included (i) scaled log-odds predicted by an ideal observer fitted from the Canadian English corpora, (ii) scaled log-odds predicted by relative frequency differences in SUBTLEX Zipf scores (Brysbaert and New, 2009), and (iii) version (sum-coded; /ɪ/ = 1, /ɛ/ = -1). Random effects included by-subject intercepts and slopes, and by-word-pair intercepts.

Listeners were biased toward /ɪ/, with an estimated baseline probability of 0.65 of /ɪ/-response according to the intercept of the regression model. Version also significantly predicted responses ($\beta = 0.54$, $z = 5.01$, $p < 0.001$), indicating that tokens morphed from /ɪ/-words were more likely to be categorized as [ɪ], and vice versa. The same regression coefficients were used in all models below to offset the effects of these variables when computing categorization responses.

3.4 Learning model

The learning model is a Bayesian ideal adaptor, as proposed in Kleinschmidt and Jaeger (2015). This belief-updating model assumes that listeners represent vowel categories as multivariate Gaussian distributions over F1 and F2, parameterized by mean μ_c and covariance Σ_c. Listeners' prior knowledge for each category c is estimated from the speech corpus and parameterized by $\mu_{c,0}$ and $\Sigma_{c,0}$. Two hyperparameters, κ_0 and ν_0, encode listeners' confidence in the prior mean and covariance, respectively. κ_0 and ν_0 can be interpreted as pseudo-counts: smaller values imply weaker prior commitment and therefore faster adaptation to incoming evidence.

Upon exposure to a talker, the prior is updated using the observed input for each category. Let N_c denote the number of observed tokens in category c. The posterior mean and covariance are given by:

$$\mu_{c,N} = \frac{K_{c,0}\,\mu_{c,0} + N_c\,\bar{x}_c}{K_{c,0} + N_c} \tag{1}$$

$$\Sigma_{c,N} = \frac{\nu_{c,0}}{\nu_{c,0}+N_c}\,\Sigma_{c,0} + \frac{N_c}{\nu_{c,0}+N_c}\Bigg(\Sigma_{x,c} + \frac{K_{c,0}}{K_{c,0}+N_c}\,(\bar{x}_c-\mu_{c,0})(\bar{x}_c-\mu_{c,0})^\top\Bigg) \tag{2}$$

Larger N_c provides stronger evidence from the exposure talker and yields posterior category representations that more strongly reflect the recent input. Note that the update of covariance (Eq. 2) is also influenced by the updating of the prior mean.

3.5 Generalization models

Building on the learning model, we implemented three strategies for how listeners might deploy exposure-updated knowledge at test.

3.5.1 Talker-specific models

Since the experiment employed the same talker during training and test, listeners may apply the exposure-updated representation directly to test tokens. This talker-specific model is a straightforward application of the ideal adaptor: the posterior distributions estimated during exposure serve as the generative model for categorizing all test items. Under this model, learning is expressed uniformly regardless of the frequency structure of the test context. It therefore serves as the null hypothesis for expression gating: lexical frequency does not modulate the expression of learning, and high- and low-frequency words rely on the same updated model.

3.5.2 Mixture-of-expectations (MoE) model

At test, listeners may have uncertainty about the applicability of this updated generative model and partially fall back on the prior knowledge, interpolating between prior knowledge and the exposure-updated representation. In this model, categorization of a test token x_g reflects a weighted combination of the posterior predictive distribution (after exposure) and the prior predictive distribution (before exposure), governed by $w \in [0,1]$. Smaller w indicates greater reliance on the prior $(\mu_{c,0}, \Sigma_{c,0})$ and larger w greater reliance on the updated model $(\mu_{c,N}, \Sigma_{c,N})$. Listeners may apply the updated knowledge differently depending on word frequency. We therefore allow w to vary by frequency condition. We used the best-fitting talker-specific model as the updated model. Each term in the mixture is the posterior predictive distribution obtained by marginalizing over the category parameters, yielding multivariate t distributions:

$$\begin{aligned} p(x_g \mid c) &= \int p(x_g \mid \theta_c)\, p(\theta_c)\, d\theta_c \\ &= w\,\mathcal{T}\Bigg(\mathbf{x}_g \mid \boldsymbol{\mu}_N, \frac{\kappa_N+1}{\kappa_N(\nu_N-D+1)}\mathbf{S}_N,\, \nu_N-D+1\Bigg) \\ &\quad + (1-w)\,\mathcal{T}\Bigg(\mathbf{x}_g \mid \boldsymbol{\mu}_0, \frac{\kappa_0+1}{\kappa_0(\nu_0-D+1)}\mathbf{S}_0,\, \nu_0-D+1\Bigg) \end{aligned} \tag{3}$$

, where D is the number of acoustic dimensions (D=2). We did not mix the two Gaussian distributions directly. Instead, we computed category likelihoods under the corresponding prior/posterior predictive functions and then re-normalized the resulting probabilities after combination. w is separately fitted for high- and low- frequency words.

3.5.3 Hierarchical Bayesian model

Rather than falling back on the prior, listeners may still adopt the updated model but deploy it with greater uncertainty. We capture this by introducing a lower-level representation during test (parameterized by μ_g, Σ_g) that is allowed to deviate from the exposure-updated higher-level model (μ_N, Σ_N). The degree of deviation is governed by the hyperparameters κ_1 and ν_1. This formulation is identical to the baseline ideal adaptor, except that no additional exposure is provided. Operationally, the best-fitting posterior distribution in the talker-specific model is taken as the higher-level model. Since listeners knew that the same talker is speaking during test, the posterior mean of μ_g and Σ_g should be centered on μ_N and Σ_N, respectively. Given that $E[\Sigma_g] = \Sigma_N$ and $E[\mu_g] = \mu_N$, the test mean (μ_g) and covariance (Σ_g) are sampled from:

$$\mu_g \sim \mathcal{N}(\mu_N, \frac{1}{\kappa_N}\Sigma_g), \quad \kappa_1 = \lambda \tag{4}$$

$$\Sigma_g \sim \mathcal{IW}(S_g, \nu_1),\ S_g = (\nu_1 - D - 1)\,\Sigma_N,\ \nu_1 = \lambda + D + 1 \tag{5}$$

,where D is the number of acoustic dimensions (D=2).

For simplicity, we use a single pooling parameter λ to represent confidence in the updated model. Note that, to ensure the covariance update is always well-defined, we set $\nu_1 = \lambda + D + 1$. The resulting posterior predictive is a multivariate student-t distribution given as

$$\mathcal{T}\left(\boldsymbol{\mu}_N, \frac{\lambda+1}{\lambda+2}\Sigma_N, \lambda+2\right) \tag{6}$$

Crucially, λ governs the confidence with which the learned representation is deployed at test, not the degree of learning itself. Larger λ yields stronger pooling, reflecting more confident deployment of the exposure-updated model; smaller λ permits greater deviation, reflecting reduced confidence in applying the updated representation to the current test context. λ is fit for high and low frequency words separately.

3.6 Optimization and model selection

Model parameters were estimated by minimizing binary cross-entropy (BCE) loss. To ensure that model fitting was sensitive to the qualitative pattern across all frequency conditions, we computed BCE on the aggregated response proportions for each group × frequency configuration × step combination and then averaged across cells. For the talker-specific model and the hierarchical model, we set the upper and lower limit of κ, ν, and λ to range between 1 and 10,000. For MoE, w ranges between 0 and 1. Both MoE and HBM were built on the best-performing talker-specific model; their transfer parameters were the only additional free parameters.

To assess how well the models recovered the signature learning patterns, we also evaluated their predictions of log-odds differences between the biasing conditions (lowering/raising) and the control condition. For each frequency configuration f, we computed model–human mismatch in the exposure–control contrasts. Let $g \in \{\text{lowering}, \text{raising}, \text{control}\}$ index exposure group, s index continuum step, and q index word pair. We first averaged response proportions within each group across participants i (Eq. 7), then computed the log-odds contrast between each exposure group and the control group (Eq. 8). Finally, we computed RMSE between model-predicted and human log-odds contrasts (Eq. 9).

$$\bar{p}_{g,f,s,q} = \frac{1}{N_{g,f}} \sum_{i=1}^{N_{g,f}} p_{i,g,f,s,q} \tag{7}$$

$$\Delta_{d,f,s,q} = \text{logit}(\bar{p}_{d,f,s,q}) - \text{logit}(\bar{p}_{\text{control},f,s,q}) \tag{8}$$

where $d \in \{\text{lowering}, \text{raising}\}$.

$$\text{RMSE} = \sqrt{\frac{1}{M} \sum_{d,f,s,q} \left(\Delta^{\text{human}}_{d,f,s,q} - \Delta^{\text{model}}_{d,f,s,q}\right)^2} \tag{9}$$

where M is the total number of exposure–control contrast × frequency-configuration × step × word-pair comparisons.

4 Simulation results

We present simulation results for the best-performing models. We evaluated model performance along two complementary dimensions. First, we report BCE in the raw probability space, which reflects global correspondence between model-predicted response probabilities and observed response proportions. Second, because the central empirical question concerns how exposure effects are expressed relative to the control condition, we computed a contrast-based RMSE over log-odds differences between each biasing condition and the corresponding control condition. Thus, BCE asks which model best captures the overall categorization pattern, whereas RMSE asks which model best captures the theoretically critical exposure-control contrast pattern. As shown in Table 1 the MoE model achieved the lowest BCE, indicating the best global probability-space fit. However, the hierarchical Bayesian model achieved the lowest contrast-based RMSE, indicating the closest recovery of the key generalization pattern. Because our primary question concerns selective expression of learning across frequency contexts, we focus below on the contrast-based comparison (Figure 2) while also reporting the global BCE metric. In the Appendix, Figure 6 and Figure 7 present model predictions in the raw probability space and predicted log-odds differences at each continuum step.

4.1 Talker-specific model

The best-performing talker-specific model ($\kappa = 27$ and $\nu = 10.1$) reproduces the main qualitative patterns in the human data (Figure 2): it predicts recalibration for /ɪ/-lowering, but not /ɛ/-raising in the HH condition, /ɪ/-lowering only in the HL condition and /ɛ/-raising only in the LH condition. These successes arise from the model's built-in

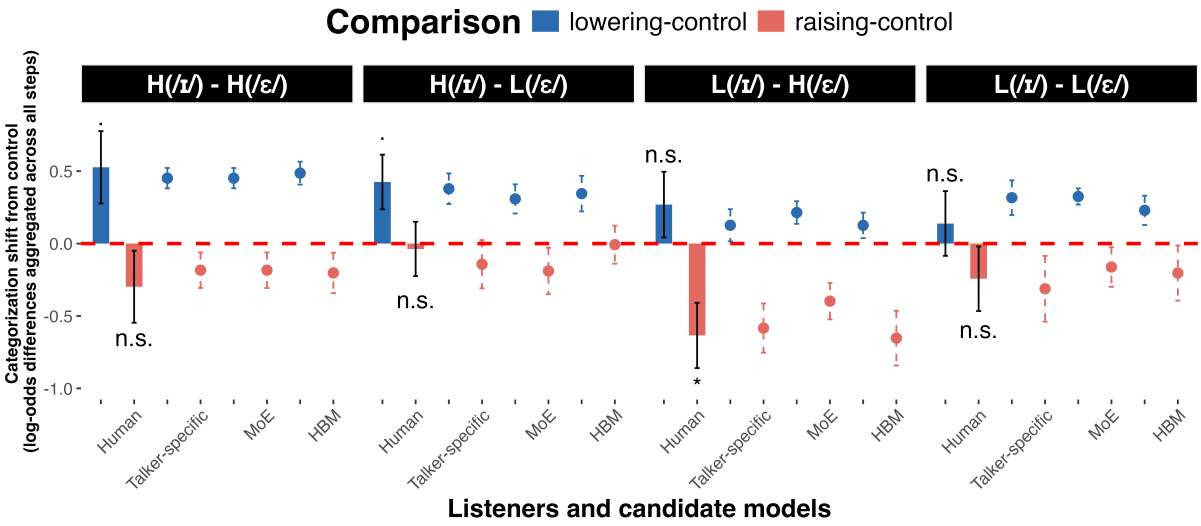


Figure 2: Log-odds differences between lowering/control and raising/control, averaged across continuum steps in each frequency condition. The red horizontal line indicates no difference between a biasing group and the control group. Positive values indicate more /ɪ/ responses than in the control group, whereas negative values indicate fewer /ɪ/ responses than in the control group. Bars indicate listeners' generalization effects. For listeners, statistical comparisons were based on post-hoc pairwise comparison derived from the linear mixed-effects model in the original experiment. P-values were corrected for multiple comparisons. Dots show the generalization effects of the best-performing models. Significance levels are coded as follows: . p < .10, * p < .05, ** p < .01, *** p < .001.

sensitivity to the acoustic properties of each test token—that is, from token-level acoustic similarity between exposure and test distributions. Importantly, the model fails to capture the LL condition: it predicts substantial learning effects in both biasing directions, whereas the corresponding effects in human data were smaller and not statistically reliable. This suggests that acoustic similarity alone accounts for most of the generalization pattern, but cannot explain why learning is not expressed when both lexical alternatives are low-frequency.

4.2 Mixture-of-expectations (MoE)

The best-performing MoE parameters were $w_h = 1$ and $w_l = 0.84$, indicating that the model relies entirely on the updated representation for high-frequency words, but places slightly more weight on the prior for low-frequency words. The MoE achieves the lowest BCE among the three models (Table 1), indicating the best global fit in probability space. However, as shown in Figure 2, the MoE model does not improve beyond the talker-specific model in capturing the theoretically critical patterns. This can also be confirmed by the largest RMSE among the three models. The model predicts a pronounced lowering effect in the LH condition, which is not observed in listeners' responses. It underestimates the degree of /ɛ/-raising in the LH condition. And crucially, it predicts robust /ɪ/-lowering in the LL condition, just as the talker-specific model does. Taken together, the MoE model predictions of learning effects do not improve beyond the learning model. The lack of improvement, particularly in LL, is likely because the empirical prior and version bias already bias the prior model toward /ɪ/, whereas listeners in LL do not exhibit an /ɪ/ bias. Consequently, lowering w_l does not improve model fit. This in turn helps explain why the best-performing model assigns a relatively higher value to w_l that preserves learning effects.

4.3 Hierarchical Bayesian model

The best performing hierarchical model parameters were $\lambda_h = 25.7$ and $\lambda_l = 17.1$, indicating higher confidence in the updated model for high-frequency words, and lower confidence for low-frequency words. As in Table 1, in global BCE, the hierarchical model performs worst among the three, indicating the worst performance among the models. The poorer predictive performance likely arises because we used a single parameter, λ, to govern uncertainty in both the posterior mean and covariance. In practice, this gives the model fewer degrees of freedom. It is possible that listeners have different degrees of confidence in the updated

estimates of the mean and covariance, and that allowing these to vary separately would yield better predictions.

Despite its higher BCE, the hierarchical model provides the best recovery of the condition-specific learning effects, as reflected in the lowest RMSE (Table 1). As shown in Figure 2, it comes closest to replicating the qualitative pattern in LL: the predicted /ɪ/-lowering effect is smaller than in the other two models, moving in the direction of the observed null effect. It also provides a better fit to the null raising effect in HL. Unlike the MoE, the hierarchical model attenuates learning not by mixing in a biased prior but by injecting additional uncertainty into the updated representation itself. This mechanism is not undermined by the /ɪ/ bias in the prior, which could be why it succeeds where the MoE does not.

We note that the hierarchical model's recovery of the LL pattern is approximate rather than exact—it reduces the predicted effect but does not fully eliminate it. The RMSE values are broadly consistent with the qualitative patterns visible in Figure 2: the hierarchical model shows the best recovery of condition-specific learning effects, while the talker-specific model and MoE perform comparably. Our goal here is not to identify the definitive account but to evaluate candidate mechanisms. The key finding is that among the three models, only the hierarchical model moves predictions in the right direction for the critical LL condition, and it does so through reduced expression confidence for low-frequency words—consistent with the expression-gating hypothesis.

Table 1: Best-fitting parameters and model performance. Learning parameters govern belief updating during exposure; transfer parameters govern expression of learning at test. MoE and HBM inherit the best-fitting learning parameters from the talker-specific model. BCE = mean binary cross-entropy (lower = better global fit). RMSE = root mean square error measuring the discrepancy between model-predicted and observed generalization effects (log-odds differences relative to control, averaged across continuum steps; lower is better)

Model	Learning	Transfer	BCE	RMSE
Talker-specific	$\kappa = 27$ $\nu = 10.1$	N/A	0.492	0.141
MoE	$\kappa = 27$ $\nu = 10.1$	$w_h = 1.00$ $w_l = 0.84$	**0.491**	0.146
HBM	$\kappa = 27$ $\nu = 10.1$	$\lambda_h = 25.7$ $\lambda_l = 17.1$	0.526	**0.121**

5 Discussions

This paper introduces a modeling framework that formalizes listeners' generalization behavior as perceptual inference under uncertainty. We applied it to LK26, a recent experiment in which listeners were exposed to shifted vowel pronunciations embedded in high-frequency words and then tested on minimal pairs varying in frequency structure (HH, HL, LH, LL). The key empirical finding is that identical learning—acquired from the same exposure items—was expressed selectively at test, depending on the frequency configuration of the test pairs.

Our modeling goal is to determine whether the talker-specific model's built-in sensitivity to acoustic similarity suffices, or whether an additional expression-gating mechanism modulated by lexical frequency is needed. We find that the talker-specific model already captures the generalization patterns in HH, HL, and LH, but overpredicts learning in LL. The hierarchical Bayesian model, which allows reduced confidence in deploying the learned representation for low-frequency words, best recovers the specific pattern across condition, as measured by the contrast-based RMSE.

Critically, the partial-transfer models, including the mixture-of-expectations and hierarchical models, do not introduce lexical frequency as an additional bias in categorization itself. Rather, building on the output of the learning model, they construct different generative models at test that vary in how tightly they are coupled to the exposure-updated representation. In this sense, these models formalize variability in the expression of learning rather than differences in the learning process itself.

Regarding model fitting and comparison, it is important to note that the models were not tailored to the observed empirical pattern. Although the MoE and HBM allowed transfer to vary by lexical frequency, the direction and magnitude of this effect were not imposed a priori. The fitted parameters could therefore have converged to no difference between high- and low-frequency contexts, or even to stronger transfer for low-frequency words. Moreover, because the talker-specific model does not include frequency-based gating, it could in principle have provided the best fit. Thus, the HBM's advantage on the contrast-based RMSE should be interpreted as an empirical outcome of

model fitting, rather than as a pattern built into the model by design. Moreover, the MoE also included frequency-specific parameters but did not best recover the contrast pattern, indicating that allowing parameters to vary by frequency was not sufficient by itself.

Our findings align with the predictions of usage-based models of phonology, which hold that listeners encode and retain fine-grained acoustic detail in memory, and that the robustness of these representations varies with frequency (Goldinger, 1996, 1998; Pierrehumbert, 2001; Wedel, 2006). Agent-based simulations of sound change have shown that—at least in some scenarios—high-frequency words tend to undergo change earlier or more rapidly than low-frequency words (Pierrehumbert, 2001; Todd et al., 2019). Our results suggest that similar stabilizing forces operate on a much shorter timescale within individual listeners: expression of short-term perceptual learning can be gated by lexical frequency, with lower-frequency words less affected by recently learned phonetic adjustments. If this perceptual conservatism is repeated over time, it may contribute to the frequency-conditioned trajectories observed in sound change. This is plausible given the structure of the lexicon: the majority of words in English are extremely rare (Mandelbrot, 1961; Piantadosi, 2014). It is sensible for listeners to treat pronunciations heard during brief exposure as relatively less relevant for predicting how such words should sound.

A related question is whether the lexical frequency of exposure items also shapes learning. LK26 did not test this, because all exposure items were high-frequency words. Recent work by Da Silva Vieira (2026) examined exposure lexical frequency and found no significant learning with low-frequency exposure items, raising the possibility that word frequency modulates the uptake of acoustic input (though see Koo et al. (2023) for contrasting evidence). However, differences in task, stimuli, and test design make direct comparison difficult. Future work should systematically cross exposure and test frequency.

Beyond the present within-talker case study, this framework can inform debates about cross-talker generalization, where competing accounts attribute transfer either to exposure to multiple talkers (Bradlow and Bent, 2008; Aoki and Zellou, 2025b) or to acoustic similarity (Xie and Myers, 2017; Xie and Jaeger, 2020; Jin et al., 2025) alone. The ideal adaptor captures both training–test similarity and cumulative exposure effects, while constrained models like MoE and HBM can isolate mechanisms beyond quantity or similarity alone.

Limitations

Several limitations should be noted. First, LK26 is, to our knowledge, the only experiment of its kind to compare generalization responses across words differing in lexical frequency. However, the different response pattern observed for LL trials may partly reflect properties of the stimulus manipulation rather than lexical frequency *per se*. In particular, the psychophysical increments between stimulus steps might not be identical across continua, as the test steps were generated through curve fitting based on categorization responses in the norming study. Moreover, both the test stimuli and the original recordings were produced by a single Canadian English speaker. It is therefore possible that the observed pattern reflects a combination of speaker-specific pronunciation, stimulus construction, and idiosyncrasies of the recruited participant sample. Further experiments are needed to determine whether the current findings can indeed be replicated.

Second, our model comparison should be interpreted with caution. We evaluated the best-fitting version of each candidate model using two complementary metrics: BCE as a measure of global fit in probability space, and contrast-based RMSE as a measure of how well each model recovered the signature exposure-control patterns. Although these comparisons are informative, differences in fit should not be taken as definitive evidence that the candidate mechanisms are uniquely identifiable from the present data. A stronger test would require simulation-based model recovery analyses, cross-validation, and related robustness checks (Wilson and Collins, 2019). Thus, the present comparison should be viewed as a theoretically motivated proof of concept rather than a definitive adjudication among mechanisms.

Acknowledgment

We thank Effie Kapnoula, Arthur Samuel, Clara Martin, Shawn Cummings, and others for their helpful comments on this work.

References

Jessica E. D. Alexander and Lynne C. Nygaard. 2019. Specificity and generalization in perceptual adapta-

tion to accented speech. *The Journal of the Acoustical Society of America*, 145(6):3382–3398.

Nicholas Aoki and Georgia Zellou. 2025a. When multiple talker exposure is necessary for cross-talker generalization: Insights into the emergence of sociolinguistic perception. *Glossa Psycholinguistics*, 4(1).

Nicholas B Aoki and Georgia Zellou. 2025b. Apparent talker variability and speaking style similarity can enhance comprehension of novel l2-accented talkers. *Language and Speech*, page 00238309251390505.

Melissa M. Baese-Berk, Ann R. Bradlow, and Beverly A. Wright. 2013. Accent-independent adaptation to foreign accented speech. *The Journal of the Acoustical Society of America*, 133(3):EL174–EL180.

Jeffrey S. Bowers, Nina Kazanina, and Nora Andermane. 2016. Spoken word identification involves accessing position invariant phoneme representations. *Journal of Memory and Language*, 87:71–83.

Ann R. Bradlow and Tessa Bent. 2008. Perceptual adaptation to non-native speech. *Cognition*, 106(2):707–729.

Marc Brysbaert and Boris New. 2009. Moving beyond kučera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american english. *Behavior Research Methods*, 41(4):977–990.

Kateřina Chládková, Václav Jonáš Podlipský, and Anastasia Chionidou. 2017. Perceptual adaptation of vowels generalizes across the phonology and does not require local context. *Journal of Experimental Psychology: Human Perception and Performance*, 43(2):414–427.

Constance M. Clarke and Merrill F. Garrett. 2004. Rapid adaptation to foreign-accented English. *The Journal of the Acoustical Society of America*, 116(6):3647–3658.

Cynthia M Connine, Debra Titone, and Jian Wang. 1993. Auditory word recognition: extrinsic and intrinsic effects of word frequency. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 19(1):81.

Marcelo Augusto Da Silva Vieira. 2026. Adaptation to communication constraints: prosodic flexibility in parkinson's disease and word frequency effect on auditory learning in young adults.

Frank Eisner and James M. Mcqueen. 2005. The specificity of perceptual learning in speech processing. *Perception & Psychophysics*, 67(2):224–238.

Frank Eisner, Alissa Melinger, and Andrea Weber. 2013. Constraints on the Transfer of Perceptual Learning in Accented Speech. *Frontiers in Psychology*, 4:148.

Stephen D Goldinger. 1996. Words and voices: episodic traces in spoken word identification and recognition memory. *Journal of experimental psychology: Learning, memory, and cognition*, 22(5):1166.

Stephen D Goldinger. 1998. Echoes of echoes? an episodic theory of lexical access. *Psychological review*, 105(2):251.

Robert D. Hawkins, Michael Franke, Michael C. Frank, Adele E. Goldberg, Kenny Smith, Thomas L. Griffiths, and Noah D. Goodman. 2023. From partners to populations: A hierarchical Bayesian account of coordination and convention. *Psychological Review*, 130(4):977–1016.

Jordan Hosier and Klinton Bicknell. 2018. Generalization in Accent Adaptation as Hierarchical Bayesian Inference.

Kaori Idemaru and Lori L. Holt. 2011. Word Recognition Reflects Dimension-based Statistical Learning. *Journal of Experimental Psychology. Human Perception and Performance*, 37(6):1939–1956.

Alexandra Jesse and James M. McQueen. 2011. Positional effects in the lexical retuning of speech perception. *Psychonomic Bulletin & Review*, 18(5):943–950.

Zhengyang Jin, Yuhao Zhu, and T Florian Jaeger. 2025. Latent speech representations learned through self-supervised learning predict listeners' generalization of adaptation across talkers. In *Proceedings of the annual meeting of the cognitive science society*, volume 47.

Charles Kemp, Andrew Perfors, and Joshua B. Tenenbaum. 2007. Learning overhypotheses with hierarchical Bayesian models. *Developmental Science*, 10(3):307–321.

Dave F. Kleinschmidt and T. Florian Jaeger. 2015. Robust speech perception: Recognize the familiar, generalize to the similar, and adapt to the novel. *Psychological review*, 122(2):148–203.

Hahn Koo, Reiko Kataoka, Julia Thomas Swan, and Christina Y Tzeng. 2023. Effects of lexical frequency on the post-exposure magnitude of recalibration in lexically guided perceptual learning: An explorative analysis. *JASA Express Letters*, 3(8).

Tanya Kraljic and Arthur G. Samuel. 2006. Generalization in perceptual learning for speech. *Psychonomic Bulletin & Review*, 13(2):262–268.

Wei Lai and Meredith Tamminga. 2024. Phonetics–phonology mapping in the generalization of perceptual learning. *Journal of Phonetics*, 103:101295.

Xinyu Leslie Liao and Yoonjung Kang. 2026. Directional bias, lexical competition, and item frequency effects in sound change: experimental evidence from perceptual learning. Submitted to Language.

Yiming Lu and Xin Xie. 2024. Modeling cue reweighting in dimension-based statistical learning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46.

Benoit Mandelbrot. 1961. On the theory of word frequencies and on related markovian models of discourse. *Structure of language and its mathematical aspects*, 12:190–219.

Jessica Maye, Richard N. Aslin, and Michael K. Tanenhaus. 2008. The Weckud Wetch of the Wast: Lexical Adaptation to a Novel Accent. *Cognitive Science*, 32(3):543–562.

Bob McMurray and Allard Jongman. 2011. What information is necessary for speech categorization? Harnessing variability in the speech signal by integrating cues computed relative to expectations. *Psychological review*, 118(2):219–246.

Holger Mitterer, Yiya Chen, and Xiaolin Zhou. 2011. Phonological Abstraction in Processing Lexical-Tone Variation: Evidence From a Learning Paradigm. *Cognitive Science*, 35(1):184–197.

Holger Mitterer and Eva Reinisch. 2017. Surface forms trump underlying representations in functional generalisations in speech perception: the case of German devoiced stops. *Language, Cognition and Neuroscience*, 32(9):1133–1147.

Kuniko Nielsen and Colin Wilson. 2008. A hierarchical bayesian model of multi-level phonetic imitation. In *Proceedings of the 27th west coast conference on formal linguistics*, pages 335–343. Cascadilla Proceedings Project Los Angeles.

D Norris, James McQueen, and Anne Cutler. 2003. Perceptual learning in speech. *Cognitive Psychology*, 47(2):204–238.

Bozena Pajak, Klinton Bicknell, and Roger Levy. 2013. A model of generalization in distributional learning of phonetic categories.

Bozena Pajak, Alex B. Fine, Dave F. Kleinschmidt, and T. Florian Jaeger. 2016. Learning Additional Languages as Hierarchical Probabilistic Inference: Insights From First Language Processing. *Language Learning*, 66(4):900–944.

Steven T. Piantadosi. 2014. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin & Review*, 21(5):1112–1130.

Janet B. Pierrehumbert. 2001. Exemplar dynamics: Word frequency, lenition and contrast. In Joan L. Bybee and Paul J. Hopper, editors, *Typological Studies in Language*, volume 45, pages 137–158. John Benjamins Publishing Company, Amsterdam.

Janet B Pierrehumbert. 2016. Phonological representation: Beyond abstract versus episodic. *Annual Review of Linguistics*, 2(1):33–52.

Eva Reinisch and Lori L. Holt. 2014. Lexically Guided Phonetic Retuning of Foreign-Accented Speech and Its Generalization. *Journal of experimental psychology. Human perception and performance*, 40(2):539–555.

Irene B R Smith, Morgan Sonderegger, and The SPADE Consortium. 2025. Patterns of pre-nasal allophony across dialects of English: A multi-corpus study of the /ɪ/-/ɛ/ contrast. OSF preprint.

Maryann Tan and T Florian Jaeger. 2025. Learning to understand an unfamiliar talker: Testing distributional learning as a model of rapid adaptive speech perception. *Cognition*, 265:106195.

Maryann Tan, Xin Xie, and T Florian Jaeger. 2021. Using rational models to interpret the results of experiments on accent adaptation. *Frontiers in Psychology*, 12:676271.

Simon Todd, Janet B. Pierrehumbert, and Jennifer Hay. 2019. Word frequency effects in sound change as a consequence of perceptual asymmetries: An exemplar-based model. *Cognition*, 185:1–20.

Andrew B Wedel. 2006. Exemplar models, evolution and language change. *The Linguistic Review*, 23(3).

Robert C Wilson and Anne GE Collins. 2019. Ten simple rules for the computational modeling of behavioral data. *elife*, 8:e49547.

Xin Xie and T. Florian Jaeger. 2020. Comparing non-native and native speech: Are L2 productions more variable? *The Journal of the Acoustical Society of America*, 147(5):3322–3347.

Xin Xie, Chigusa Kurumada, and T. Florian Jaeger. 2023. What we do (not) know about the mechanisms underlying adaptive speech perception: A computational framework and review. *Cortex*, 166:377–424.

Xin Xie, Linda Liu, and T Florian Jaeger. 2021. Cross-talker generalization in the perception of nonnative speech: A large-scale replication. *Journal of Experimental Psychology: General*, 150(11):e22.

Xin Xie and Emily B. Myers. 2017. Learning a talker or learning an accent: Acoustic similarity constrains generalization of foreign accent adaptation to new talkers. *Journal of Memory and Language*, 97:30–46.

Weijie Xu and Richard Futrell. 2024. A hierarchical Bayesian model for syntactic priming. *arXiv preprint*. ArXiv:2405.15964 [cs].

A Additional figures

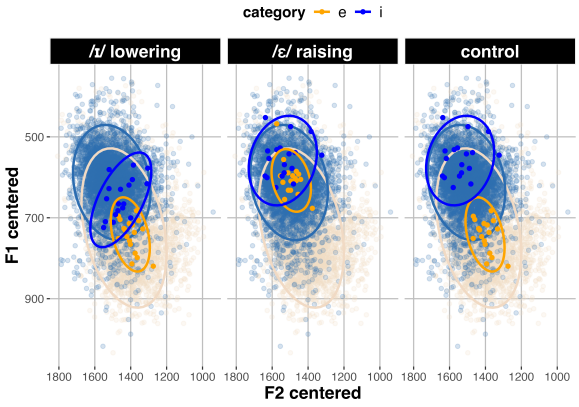


Figure 3: Distribution of exposure trials (shown in solid dots), against the Canadian corpus (in faded dots)

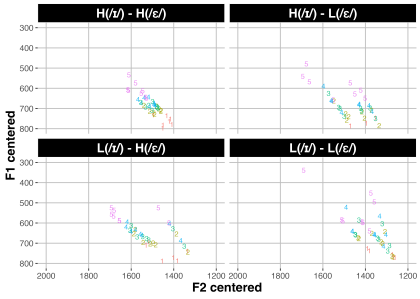


Figure 4: Test items across the four frequency combinations. Values 1–5 indicate the proportion of /ɪ/ responses estimated in a pilot norming study, corresponding to 0.01, 0.25, 0.50, 0.75, and 0.99, respectively.

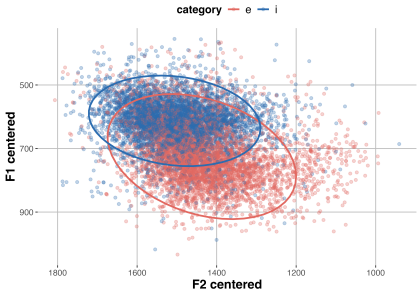


Figure 5: The Canadian corpus. Each dot represents an item containing /ɪ/ or /ɛ/. F1 and F2 were centered relative to each talker's cue mean, and the grand mean across all talkers was then added back. Ellipses indicate the 95% confidence region of the Gaussian distributions fitted across all talkers..

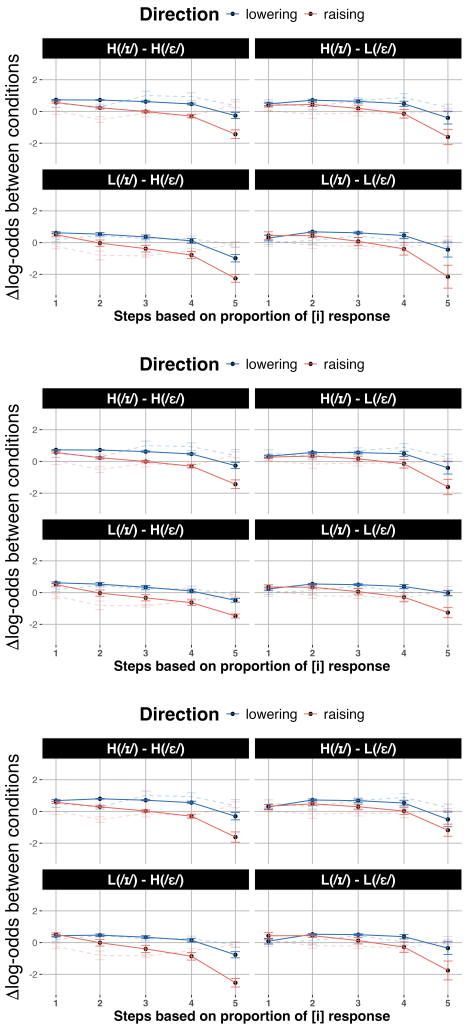


Figure 7: Predictions of the best-performing talker-specific model (top), mixture of expectation model (middle), and hierarchical Bayesian model (bottom) in the log-odds space. To highlight learning, the x-axis shows continuum steps and the y-axis shows log-odds differences between /ɪ/-lowering and control and between /ɛ/-raising and control. Solid lines show model predictions, and faded lines reproduce the human data for comparison. Error bars indicate ±1 standard error of the mean.

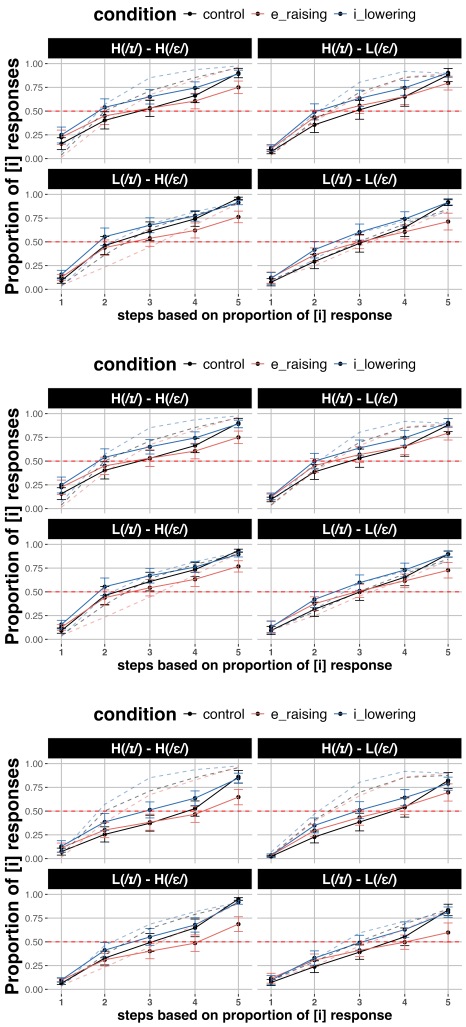


Figure 6: Predictions of the best-performing talker-specific model (left), mixture-of-expectations model (middle), and hierarchical Bayesian model (right) in the raw probability space. Solid lines show model predictions, while faded dashed lines show mean proportion of /ɪ/ responses. To improve visual clarity, only the model error bars (± 1 standard error of the mean) are shown here.

Output Languages of Strictly Local Phonological Processes

Anton Hampe
Department of Linguistics
Yale University
anton.hampe@yale.edu

Abstract

This paper investigates the relationship between strictly local phonological processes and strictly local phonotactic constraints. On the theoretical side, I identify phonological rewrite rules that do not produce strictly local output languages and that do not weakly preserve the class of strictly local languages. Empirically, I find that strictly local rules without strictly local output languages are largely absent from the PBase database.

1 Introduction

This work refines the computational typology of phonological rewrite rules. I assume basic familiarity with the mechanics of phonological rewrite rules and elementary automata theory. I recommend Hayes (2011) for readers unfamiliar with phonological rewrite rules and Hopcroft et al. (2001) for readers unfamiliar with automata and transducers. Phonological rewrite rules have a long history in linguistics, going as far back as the grammars of Pāṇini from the 5th century BCE (Kiparsky, 2022). Today, they are often referred to as SPE-style rules, following Chomsky and Halle (1968). SPE-style rules stand in a special relationship to the regular languages, which are the string sets recognized by finite state acceptors, and the rational relations, which are the string to string maps computable by finite state transducers. The nature and history of that relationship is summarized below. Johnson (1972) noted that both the schema for morpheme structure used at the time and the environments targeted by phonological rewrite rules specify regular sets of strings, concluding that phonological surface patterns have to belong to the regular class as well. Kaplan and Kay (1994) made this result more rigorous and showed that all SPE-style rules and their common modes of application produce rational relations and regular surface languages. The optimality-theoretic grammar formalisms that are commonly used today generally do not share this property (Lamont, 2021a,b). However, typological work studying the extensions of phonological processes and surface languages suggests that these entities themselves have these properties, independent of the grammar formalisms used to model them (Heinz, 2018).

(Mielke, 2008) compiled PBase, a large database of rewrite rules used in phonological grammars. The existence of such a large database makes refining the computational properties of classes of rewrite rules particularly useful for typological work.

2 Strictly local languages

As computational phonology has matured, the early results suggesting that phonological surface patterns are regular have been refined to more restrictive, subregular classes (Heinz, 2018). One linguistically very useful class of subregular languages is the class of strictly local (SL) languages introduced by McNaughton and Papert (1971). Edlefsen et al. (2008) found that about 75% of stress patterns cataloged in the stresstyp database (Goedemans et al., 2015) belonged to the SL class. The class of SL languages is very robust, in the sense that it permits multiple equivalent definitions in terms of structural restrictions on finite-state automata, logic, closure properties, and algebraic properties (McNaughton, 1974; Rogers et al., 2012; Rogers and Pullum, 2011; Lambert, 2025).

For the purposes of this paper the characterizations in terms of automata and the characterization in terms of logical constraints are the most relevant. Each strictly local language has a canonical acceptor, such that each state is uniquely labeled by a suffix of length $n \geq 0$ of all paths leading to it. If such an acceptor exists, then the language is $(n+1)$-SL. For instance, a machine with a single state labeled by the empty string is 1-SL. Figure 1 shows

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 542–548
July 3-4, 2026 ©2026 Association for Computational Linguistics

the canonical automaton recognizing the 2-SL language $a\{a,b,c\}^*b \cup c\{a,b,c\}^*b$. This language contains all strings over the alphabet $\{a,b,c\}$ that end with b, but do not begin with b. The $\rtimes$ and $\ltimes$ symbols are special markers for the left and right boundary of the string, respectively. These markers are assumed to always mark the boundaries of the string, but are not included in the alphabet of string-internal symbols. In the context of phonology, the symbol # often plays a similar role in marking both the left and right boundaries of a word.

Alternatively, a strictly local language can be characterized in terms of a finite set of allowed or prohibited substrings, also known as n-grams. The language of the automaton in Figure 1 can be characterized as all possible strings containing only length-two substrings or bigrams from the set $\{\rtimes a, \rtimes c, aa, ab, ac, ba, bb, bc, ca, cb, cc, b\ltimes\}$. Alternatively, it can be characterized as the set of all strings over the alphabet $\{a,b,c\}$ not containing the substrings or n-grams $\{\rtimes\ltimes, \rtimes b, a\ltimes, c\ltimes\}$. In the context of phonology, an n-gram like ab being absent from a language is often phrased as the language having the constraint $*ab$, or satisfying the constraint $*ab$.

3 Strictly local functions

Just as string sets can be classified into subregular classes, string transductions can be classified into subrational classes. Subclasses of the rational relations are often named for subregular classes of languages that stand in a special relationship to them. For this paper the input strictly local (ISL) and output strictly local (OSL) classes introduced by Chandlee (2014) are of particular relevance. These classes are named for the fact that they can be structurally defined using canonical transducers that closely mirror the structure of the canonical acceptors underlying SL languages. Canonical ISL transducers have each state uniquely labeled with a suffix of length $n \geq 0$ of the input projection of all paths leading to it. Similarly, canonical OSL transducers have each state uniquely labeled by a suffix of length $n \geq 0$ of the output projection of all paths leading to it. Both classes of transductions have connections to natural classes of phonological rules. ISL functions include phonological rewrite rules $A \rightarrow B / C__D$ under simultaneous application, where A and B consist of a single letter or the empty string and C and D represent strings of bounded length. OSL functions include phonological rewrite rules $A \rightarrow B / C__D$ under iterative application, where A and B consist of a single letter or the empty string and C and D represent strings of bounded length. Processes which can be represented in both ways are included in both classes.

4 Output languages

Belonging to the ISL or OSL classes is not the only way in which a transduction may relate or share properties with the SL languages. The output language of a transduction is the set of strings the transduction produces when fed arbitrary input. In other words, it is the image of Σ^*, where Σ is the input alphabet for the transduction. Given the prevalence of SL phonotactic surface patterns, one might expect the ISL and OSL phonological rules enforcing these surface patterns to have SL output languages. For instance, final devoicing is a phonological process active in Turkish, Russian, German and many other languages. It is commonly modeled by a rewrite rule of the form $\{+voice, -sonorant\} \rightarrow \{-voice\} / __\#$. This rewrite rule produces a strictly local output language with the singular constraint $*\{+voice, -sonorant\}\#$, which makes sense since the fact that such strings are missing from the surface phonotactics of these languages is precisely what motivated the rule in the first place.

Chandlee et al. (2014) show that this is not a trivial property. Belonging to the ISL class, OSL class, or both is not enough to ensure that the output language of a transduction is strictly local. However, so far no theoretical work had been conducted checking whether simplex ISL or OSL processes which can be represented as single rewrite rules have SL output languages.

5 SL-preservation

A class of transductions weakly preserves a class of languages if the transduction produces an image that belongs to the class, for each set of inputs that belong to the class (Hashiguchi and Honda, 1976). For example, a function is weakly SL-preserving if it maps each SL subset of its domain to a SL image. Since the free monoid Σ^* forms a strictly local language, being weakly SL-preserving entails having a SL output language.

I am not aware of any decision procedure for weak SL-preservation, and to my knowledge, the work of Chandlee et al. (2014) is the only work that touches on this property. The question of whether

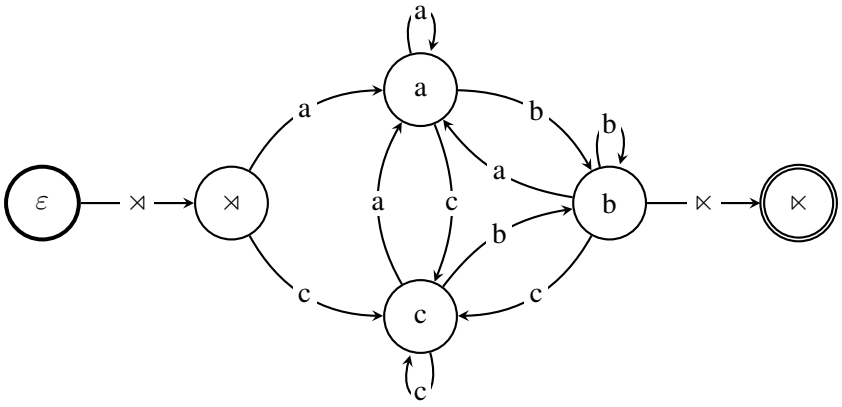


Figure 1: The canonical 2-SL automaton recognizing $a\{a,b,c\}^*b \cup c\{a,b,c\}^*b$. The initial state is marked with a thick outline, and the final state is marked with a double outline.

any ISL or OSL functions representable as simplex phonological rules do (not) weakly preserve SL languages has not been previously explored.

This work offers no decision procedure for SL-preservation either. Instead, I introduce a method which produces counterexamples to SL-preservation for a large number of SPE-style rules. The counterexamples involve creating a strictly local language which consists of alternations of contexts in which the rule applies and surface strings which result from such an application. For example, the rule $A \rightarrow B / C__$ maps the language $(CACB)^+$ to $(CBCB)^+$, regardless of whether it is applied iteratively left-to-right, right-to-left, or simultaneously. The input language is 3-SL and generated by the positive grammar $\{\rtimes CA, CAC, ACB, CBC, BCA, CB\ltimes\}$. The output language is equivalent to $(CB)^{2n}$ and thus not star-free.

This construction is not universal. It fails for rules such as $A \rightarrow B / A__$. Here, the language $(AAAB)^+$ is mapped to $(ABBB)^+$ when the rule is applied iteratively right-to-left or simultaneously. Since the image is a SL language, this is not a valid counterexample. This can be remedied by introducing a distinct symbol that separates the two strings, as in $(AACABC)^+$. The image of that language is then $(ABCABC)^*$, which is not star-free. This construction works, as long as there is some symbol C in the input alphabet which is not implicated in the rule in any way. As I have shown, a substantial subset of SPE-style rules does not weakly preserve the SL class. This means that the property of having a SL output language is not necessarily preserved under composition. Let f and g be two SPE-rules that produce strictly local output languages, but which do not weakly preserve the full SL class. The function f may map the set of all possible input strings to a strictly local language L different from Σ^*. Because g does not weakly preserve the SL class, we have no guarantee that it maps L to a SL language. The composition $g \circ f$ would then map Σ^* to a non-SL language. I return to this issue in section 6.4.

6 Theoretical results

6.1 Methodology

Hulden (2009) introduced the FOMA software package for creating finite-state machines in the machine-readable .att format. I generated .att files for all rewrite rules over the alphabet $\Sigma = \{A, B, C, D, E, F\}$ with context length of up to 2 on either side of the target. Target and change of each rule had to be distinct from each other and were restricted to be either the empty string or a single letter. The inclusion of the empty string allows for epenthesis and deletion rules. The rule involving the largest number of distinct symbols was thus $A \rightarrow B / CD__EF$. For each rule, I created automata under simultaneous application (guaranteeing membership in the ISL class) and left-to-right iterated application (guaranteeing membership in the OSL class). The resulting .att files were analyzed using the LTK software package of Lambert (2024). For this paper, I treat rules that differ only in symbol names as identical. For example, the rules $A \rightarrow B / CC__$, and $A \rightarrow B / DD__$ are treated as the same rule.

$\emptyset \rightarrow x / x__x$	$\emptyset \rightarrow x / x__xx$	$\emptyset \rightarrow x / x__$	$\emptyset \rightarrow x / yy__$
$\emptyset \rightarrow x / xy__$	$\emptyset \rightarrow x / xx__x$	$\emptyset \rightarrow x / xx__xx$	$\emptyset \rightarrow x / xx__$
$\emptyset \rightarrow x / __x$	$\emptyset \rightarrow x / __yy$	$\emptyset \rightarrow x / __yx$	$\emptyset \rightarrow x / __xx$

Figure 2: Rules producing non-SL output languages under simultaneous application.

6.2 Simultaneous application

I found that under simultaneous application there are indeed simplex SPE-style rules that produce non-SL output languages. The set of such rules is shown in Figure 2.

Notably, all such rules were epenthesis rules. I would also like to note that many of these rules are intuitively strange or phonologically atypical. Consonant epenthesis tends to be trigerred by adjacent vowels, and vowel epenthesis tends to be trigerred by illicit consonant clusters (Blevins, 2008; Hall, 2011). However, most of the rules in Figure 2 appear to copy an element from their environment. A third interesting point is that all these rules not only produce output languages which are non-SL, they produce output languages which are non-star free. The rule $\emptyset \rightarrow x / x__$ for instance produces an output language whose intersection with x^* is x^{2n}, i.e. strings of xs with an even length. This is because it maps each string x^n to the string x^{2n}. Since SL languages are closed under intersection, this makes the output language non-SL.

6.3 Iterative application

I found that there are also rules which produce non-SL output languages under iterated left-to-right application. This set of rules is given in Figure 3.

Notably, this set is a subset of the rules with non-SL output languages under simultaneous application. The rules that produce SL output languages under iterated left-to-right application, but not under simultaneous application are rendered with strike-through. When opting for right-to-left application, the rules that lose the property of non-SL output languages are the mirror images $\emptyset \rightarrow x / __yy$ and $\emptyset \rightarrow x / __yx$.

6.4 Composition

Because of the combinatorics of composing every rule with a SL output language with every other rule with a SL output languages, composition of these rules was not tested exhaustively. When randomly sampling rules and composing them, I found that the property of having a SL output language is not always preserved under composition.

The instances of two composed rules forming a complex process with a non-SL output language had a predictable relationship to the simplex rules with non-SL output languages. All of them included an initial step of epenthesis followed by a step of erasing distinctions between triggers and targets. For instance, applying the rule $\emptyset \rightarrow A / B__$ maps strings of the form B^n to strings of the form $(BA)^n$. Following this application with the rule $A \rightarrow B / B__$ in turn maps strings of the form $(BA)^n$ to strings of the form $(BB)^n$. Because the output language of the first rule lacks strings of the form B^+, and the second conspires with the first to add all strings of the form B^{2n}, the overall effect is a non-star-free output language. Neither rule on it's own produces a non-SL output language.

7 Empirical results

7.1 Methodology

I searched PBase for rules matching the template in Figure 2. Because I used concrete individual symbols in my analysis of phonological rules, but PBase (and phonologists generally) use natural classes of segments, I decided a rule matched my template if it fit my template for any subset of the alphabet. For example, a rule $\emptyset \rightarrow j / \{i, j\}__$ would count as matching the template $\emptyset \rightarrow x / x__$, even though $\{i\}$ and $\{i, j\}$ are different sets.

7.2 Potential matches

The search yielded nine matches from seven languages.[1] Eight turned out to be faulty data entry, meaning the prose description of the rule did not match the way the rule was coded. For instance, the database included a rule for glide insertion in Chrau of the form:

$\emptyset \rightarrow$ j/{g, c, b, l, d, h, k, j, m, ʃ, n, p, ɲ, t, w, ʔd, r, ʔb, ŋ, ɟ, ʔ}{c, ɟ, ʃ, ɟ, j}__

[1]The IDs of the matching rules are 595, 2000, 2320, 2349, 2558, 3030, 3225, 6261, and 6669 from the languages Agarabi, Burmese, Chrau, Czech, Eastern Ojibwa, Kiowa, and Sie.

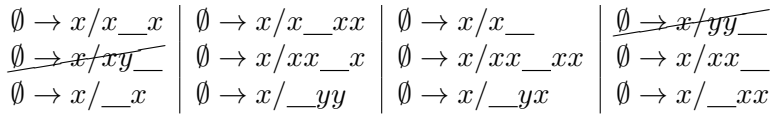


Figure 3: Rules producing non-SL output languages under left-to-right iterated application. Rules rendered with strike-through produce non-SL output languages under simultaneous application, but not under left-to-right application.

However, the original grammar by Thomas (1971) contains the following description of the rule:

"When a palatal (ch, j, nh, y, s) is in C_f position, there is usually a phonetic y-glide between the vowel and the consonant."

This description implies that a vowel should be included in the right context. Including such a vowel prevents the rule from matching the templates $\emptyset \rightarrow x/xx__$ and $\emptyset \rightarrow x/xy__$.

The ninth match in the database is the rule from Kiowa below.

$\emptyset \rightarrow$ j/{j}{h,ʔ}__

The original grammar of Watkins (1980) describes the rule as follows.

"The palatal glide spreads across the laryngeals yielding a glide onset, a brief moment of coarticulation and a glide release. The rule operates across word boundaries as well as word internally."

This description does not force the glide to be followed by a vowel. The rule as described matches the template $\emptyset \rightarrow x/xy__$. This rule template produces a non-SL output language under simultaneous application, but not under left-to-right or right-to-left application.

Let us take a closer look at how the rule $\emptyset \rightarrow x/xy__$ produces a non-SL output language under simultaneous application. The crucial surface forms are strings of the form $(xy)^n$. The rule template will only produce such strings for even n. Take the output string $xyxy$ and let us underline all the segments that must have been present in the input. Since the rule does not create any new y segments, all the instances of y must have been underlying in the input, giving us $\underline{x}\underline{y}x\underline{y}$. Since the first x could not have been preceded by the correct context for epenthesis, it too must have been present in the input. This leaves us with $\underline{xy}\underline{xy}$. The last remaining x has the context for insertion underlyingly adjacent. Therefore, the last x must have been epenthesized by the rule.

Now if we append another xy to that output string, we get $\underline{xy}\underline{xy}\underline{xy}$. The y we appended must have been present in the input for the same reason as the other ys. Namely that the phonological rule in question cannot epenthesize a y. The x must have been present in the input, since it is not preceded by the context that would trigger epenthesis.

However, this poses a dilemma. We have established that the final xy in the output must have been underlying. But if it were underlying, it should have triggered the rule and there should be another final x epenthesized at the end. This contradiction shows that $xyxyxy$ cannot be in the output language of the rule. The same argument holds for all $(xy)^n$ for odd n. Patterns such as this one that involve modulo counting fall outside the class of strictly local languages.

In terms of the Kiowa rule, this means one would need repeated instances of jhʔ or jʔh in underlying forms to get similar behavior. The grammar does not contain such underlying forms featuring repeated iterations of laryngeals and glides. This means that the rule could potentially be modified to include another type of segment, such as a vowel, in its context. Given these possibilities for reanalysis and the fact that the rule produces a SL output language when applied left-to-right (as might be expected of a spreading process that proceeds to the right), PBase supports the conclusion that strictly local phonological processes produce strictly local output languages to a striking degree.

8 Conclusion

This work identified the set of simplex phonological rules belonging to the ISL and OSL classes with context length up to 2 which produce non-SL output languages. All such rules were epenthesis rules with highly unusual contexts. A search of the PBase database produced no rules that produce non-SL output languages under iterated left-to-right ap-

plication. The database does contain a rule that would produce a non-SL output language under simultaneous application, if taken at face value. This paper also showed that the property of weakly preserving SL languages is not one that holds for a large majority of simplex ISL and OSL rewrite rules, including very common rule templates.

This second fact makes the property of producing SL output languages unstable under composition. One rule can produce an output language for which a following rule does not preserve strict locality. Understanding under which conditions rules that individually produce SL output languages may compose to yield a complex process which does not is a possible direction for future research. Empirical work investigating whether such combinations of rules exist in analyses of natural phonological systems is another possible direction.

Interestingly, all ISL and OSL process without SL output languages identified in this paper had output languages outside the star-free class. It stands to reason that ordered rewrite rule analyses enforcing counting constraints would have been noticed by now, since the notion that phonology does not count has long been a core assumption of phonology.

If it is the case that the dichotomy between SL output languages and non-star-free output languages is not an accident, that opens up interesting possibilities for restrictions on the ordering of simple phonological rules that might be of use for the study of opacity and rule interaction.

If it is the case that morpheme structure constraints exist, the theoretical portion of this paper implies restrictions on such constraints. Even if constraints on underlying representations were merely SL, a bad combination of morpheme structure constraints and simple phonological processes could conspire to produce non-star-free surface languages.

A final possible application of these results is the design of learning algorithms. Although these results eliminate relatively few simplex processes, they do add a new typologically supported bias that a learner could exploit.

References

Juliette Blevins. 2008. Consonant epenthesis: Natural and unnatural histories. *Language universals and language change*, pages 79–107.

Jane Chandlee. 2014. *Strictly local phonological processes*. University of Delaware.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2014. Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics*, 2:491–504.

Noam Chomsky and Morris Halle. 1968. *The Sound Pattern of English*. Harper & Row, New York.

Matt Edlefsen, Dylan Leeman, Nathan Myers, Nathaniel Smith, Molly Visscher, and David Wellcome. 2008. Deciding strictly local (sl) languages. In *Proceedings of the midstates conference for undergraduate research in computer science and mathematics*, volume 6, pages 6–75. Citeseer.

Rob Goedemans, Jeffrey Heinz, and Harry van der Hulst. 2015. StressTyp2, version 1. Web download archive. Available at http://st2.ullet.net.

Nancy Hall. 2011. Vowel epenthesis. *The Blackwell companion to phonology*, pages 1–21.

Kosaburo Hashiguchi and Namio Honda. 1976. Properties of code events and homomorphisms over regular events. *Journal of Computer and System Sciences*, 12(3):352–367.

Bruce P Hayes. 2011. *Introductory phonology*. John Wiley & Sons.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. *Phonological typology, phonetics and phonology*, 23:126–195.

John E Hopcroft, Rajeev Motwani, and Jeffrey D Ullman. 2001. Introduction to automata theory, languages, and computation. *Acm Sigact News*, 32(1):60–65.

Mans Hulden. 2009. Foma: a finite-state compiler and library. In *Proceedings of the Demonstrations Session at EACL 2009*, pages 29–32, Athens, Greece. Association for Computational Linguistics.

C. Douglas Johnson. 1972. *Formal Aspects of Phonological Description*. Number 3 in Monographs on Linguistic Analysis. Mouton, The Hague.

Ronald M Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational linguistics*, 20(3):331–378.

Paul Kiparsky. 2022. Pāṇini. In *The Oxford History of Phonology*. Oxford University Press.

Dakotah Lambert. 2024. System description: A theorem-prover for subregular systems: The language toolkit and its interpreter, plebby. In *Functional and Logic Programming*, pages 311–328, Singapore. Springer Nature Singapore.

Dakotah Lambert. 2025. On the structure of sets testable in the strict sense. In *Proceedings of the 16th Meeting on the Mathematics of Language (MOL 2025)*. To appear.

Andrew Lamont. 2021a. Optimality theory implements complex functions with simple constraints. *Phonology*, 38(4):729–740.

Andrew Lamont. 2021b. Optimizing over subsequences generates context-sensitive languages. *Transactions of the Association for Computational Linguistics*, 9:528–537.

Robert McNaughton. 1974. Algebraic decision procedures for local testability. *Mathematical systems theory*, 8(1):60–76.

Robert McNaughton and Seymour A Papert. 1971. *Counter-Free Automata (MIT research monograph no. 65)*. The MIT Press.

Jeff Mielke. 2008. The emergence of distinctive features.

James Rogers, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel. 2012. Cognitive and sub-regular complexity. In *International Conference on Formal Grammar*, pages 90–108. Springer.

James Rogers and Geoffrey K Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information*, 20(3):329–342.

David D Thomas. 1971. Chrau grammar. *Oceanic Linguistics Special Publications*, (7):i–258.

Laurel Jayne Watkins. 1980. *A grammar of Kiowa*. University of Kansas.

Phonological Processes as Modal Transductions

Tatevik Yolyan[1,2], **Jesse Comer**[1]
[1]University of Pennsylvania,
[2]Rutgers University, New Brunswick
Correspondence: tatevik.yolyan@rutgers.edu

Abstract

This paper argues in favor of a fundamentally new perspective on phonology via *modal logic*. We show that the class of total *Boolean Monadic Recursive Schemes* (**BMRS**), used in computational modeling of phonological processes (Bhaskar et al., 2020; Chandlee and Jardine, 2021), is equivalent in expressive power to the well-studied *modal μ-calculus*. As a corollary of this result, we obtain an alternative proof that order-preserving **BMRS** transductions capture the class of rational functions, which have been posited as a complexity bound on natural language phonological grammars.

1 Introduction

Rational functions are word-to-word maps which are realized by non-deterministic one-way finite-state transducers (FSTs).[1] Within phonological theory, the Rational Hypothesis argues that phonological maps in natural language fall within the restrictive class of rational functions (Kaplan and Kay, 1994; Johnson, 1972). We take the Rational Hypothesis as the starting point for this paper, and present a new formal perspective on rational functions and consequently, phonological maps. The conceptual map for this paper is presented in Figure 1; the relationship between phonological maps and finite state transducers that is central to the Rational Hypothesis constitutes the edge labeled (a).

(1) **The Rational Hypothesis:** Natural language phonological maps are rational.

The Rational Hypothesis provides a formal characterization of what kinds of *computations* are attested in natural language phonological grammars, and moreover predicts what kinds of computations *cannot* be realized as a phonological map in *any* natural language.

[1]In-depth discussion of rational functions can be found in (Berstel, 2007) and (Filiot, 2015).

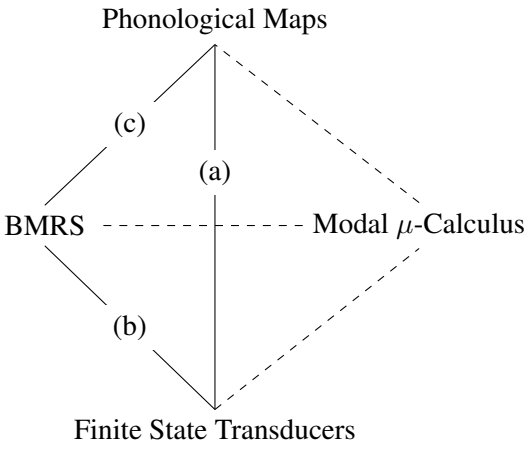


Figure 1: Conceptual map of this paper. (a): Kaplan and Kay (1994); Johnson (1972); (b): Bhaskar et al. (2020, 2023); (c): Chandlee and Jardine (2021); Yolyan (2025b,c). This paper discusses the relationships labeled with dashed lines.

It is worth noting that the rational relations are often referred to as 'regular' in computational phonology literature, and the Rational Hypothesis is better-known as the **Subregular Hypothesis** (e.g. Heinz, 2018). Though the terms 'rational' and 'regular' get conflated, it is important to keep the following distinction in mind: the *regular* relations are those which can be expressed with a *two-way* non-deterministic FST, and are therefore more expressive than the rational functions.

In recent work, Bhaskar et al. (2020, 2023) introduced Boolean Monadic Recursive Schemes (BMRS), a programming language interpreted over words, as a *logical characterization* of the rational functions. More specifically, they show that the so-called *well-defined, order-preserving* **BMRS** interpretations capture precisely the class of rational functions on strings. Their work constitutes the edge labeled (b) in Figure 1. Chandlee and Jardine (2021) present **BMRS** as an alternative framework for representing phonologi-

Proceedings of the Society for Computation in Linguistics (SCiL) 2026, pages 549–559
July 3-4, 2026 ©2026 Association for Computational Linguistics

cal maps because **BMRS** programs can be used to formally represent both the computational nature of phonological maps as well as relevant *phonological generalizations*. Subsequent work arguing for the merits of **BMRS** for phonological theory is also found in (Yolyan, 2025b,c). This relationship constitutes edge (c) of Figure 1.

This paper presents a *modal logic perspective* on rational transductions and phonological maps. More specifically, we present the modal μ-calculus, which is an extension of propositional modal logic with a *fixed point operator*, which permits recursion and self-reference in formulas (Bradfield and Stirling, 2006). There are a few motivating factors for pursuing a modal perspective. First, the modal μ-calculus is a logic with a well-understood relationship to automata and formal languages. A detailed discussion of the relationship between the modal μ-calculus and alternating tree automata can be found in Emerson and Jutla (1991); Wilke (2001). In the setting of words, the modal perspective provides an alternative logical framework for expressing phonological transductions. Second, the μ-calculus employs a least fixed point semantics for evaluating recursive formulas. Recursive formulas are also used in **BMRS** to express long-distance and iterative string maps. Establishing a connection between the semantics of these two separate formalisms means that existing results on the formal properties of the modal μ-calculus can provide new insights into the **BMRS** framework. Third, the modal perspective may provide new insights on computational learning of phonological maps. A significant consequence of the Rational Hypothesis is that it restricts the learning space for phonological maps. This restricted space has led to advances in learning FST representations of rational functions, such as Chandlee (2014); Jardine et al. (2014); Oncina et al. (1993). More recent work from Yolyan (2025a) presents an approach to learning phonological maps represented as **BMRS** programs. There have also been recent advances on learning μ-calculus formulas, such as Koudijs (2022); ten Cate and Koudijs (2024). A formal connection between modal formulas and **BMRS** programs may ultimately lead to new insights that are valuable for learning.

With these motivating factors in mind, this paper makes several contributions which make up the three dotted lines in Figure 1. The main result of this paper is that the modal μ-calculus and total **BMRS** define the same unary properties over words. More precisely, for each output predicate of a total **BMRS** program, there is a modal μ-calculus formula which is true at all and only indices in a word for which that output predicate is true. An immediate consequence of our main result is that it is possible to represent rational functions as order-preserving interpretations of μ-calculus formulas, which have not been pursued in previous modal logic literature. This paper can moreover be viewed as an extension of previous work by Graf (2010), who uses modal logic to express well-formedness constraints on surface forms in phonology. The fixed point operator in the modal μ-calculus is the crucial formal component that makes it possible to lift modal logic formulas that represent surface form constraints to modal μ-calculus formulas which, given the formal results of this paper, represent phonological maps. An additional potential contribution of this paper is that it may be of interest to modal logicians, as it provides a new way to express modal μ-calculus expressions and connects modal logic to programming languages.

The structure of this paper is as follows. Section 2 gives a high-level presentation of how FSTs, **BMRS**, and modal logic are used to represent phonological maps. This section provides the motivation for this work, and the intuition for why a modal logic perspective is warranted. A more formal presentation of **BMRS** and the modal μ-calculus follow in Sections 3 and 4. The main result of this paper is presented in Section 5, followed by concluding remarks in Section 6.

2 Phonological Transductions

This section presents two examples of phonological maps to illustrate how FSTs, **BMRS**, and modal logic can be used to represent phonological maps. We present first a non-recursive example in (2) in order to provide an informal introduction to the three frameworks. We then consider an iterative process in (5) to demonstrate the evaluation of recursive predicates. Formal details of BMRS and modal logic follow in the subsequent two sections.

Consider the vowel nasalization map in (2), in which a vowel becomes [+nasal] when it precedes a nasal sound. The FST representation of this map is presented in Figure 2 with features N for nasals and V for vowels. The three states of this FST represent three possible sounds in the input. For ex-

ample, the machine transitions to the state labeled $+-$ when it reads an input that is [+V,−N]. The FST reads the input from right to left. The highlighted transition [+V,−N]:[+V,+N] from state $+-$ to $-+$ represents the fact that a [−N] vowel becomes [+N] when the previous sound that was read in the input was [+N].

(2) Vowel Nasalization
V → [+nasal] /__[+nasal].

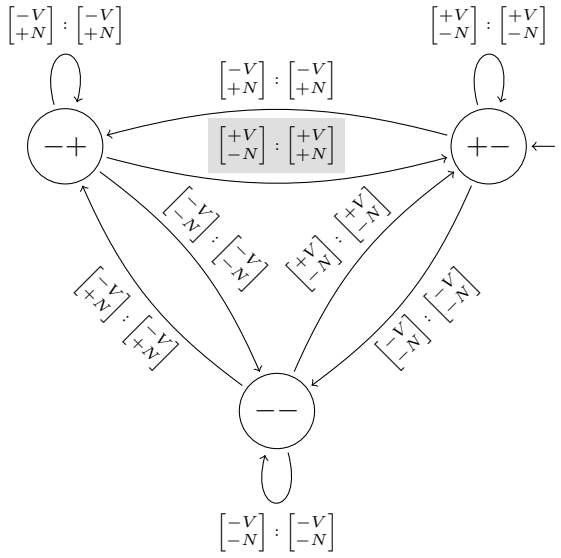


Figure 2: FST representation of vowel nasalization, where V and N represent vowel and nasal sounds.

The **BMRS** formalism provides a framework for expressing phonological maps as quantifier-free logical transductions. These transductions operate on relational structures called *words*.

Definition 1. *Given a set of unary predicate symbols Σ (called a* signature*), a Σ-word is a structure $\boldsymbol{w} = \langle w, s^w, p^w, (\sigma^w)_{\sigma\in\Sigma}\rangle$, where w is a finite set of indices; s^w and p^w denote (partial) successor and predecessor functions on w, undefined only at the last and first indices, respectively; and each σ^w is a subset of w. A Σ-string is a Σ-word with exactly one predicate true at each index. We write $\boldsymbol{Words}_\Sigma$ to denote the class of Σ-words.*

When logical transductions are adapted for representing phonological maps, the unary predicates in the word represent phonological features. Within the BMRS framework in particular, the predicates σ^w are represented with Boolean-valued predicates such that $\sigma^w(x) = \top$ if and only if $x \in \sigma^w$. Logical transductions represent maps in the following way: for every predicate σ^w of a word **w**, the logical transduction defines a new word **w′** with predicates σ' (called *output predicates*). The BMRS representation of vowel nasalization is given in (3) with predicates N and V, for nasal and vowel sounds. The two output predicates N' and V' define an output string by specifying which indices of the output string are nasals and vowels, respectively.

(3) BMRS representation of vowel nasalization
$V'(x) \quad = V(x)$
$N'(x) \quad =$ **if** $V(x)$ **then** $N(\mathrm{s}(x))$ **else** $N(x)$[2]

Note that **BMRS** programs are expressed using a `if...then...else` syntax. The output predicate $N'(x)$ is evaluated as follows. For every index x, if x is a vowel in the input word, then x is [+N] in the output word if and only if index $x+1$ of the input word is [+N]. Otherwise, x is [+N] in the output word if and only if it is [+N] in the input word. Consider an underlying-surface form pair of words (/bæn/,[bæ̃n]). This input-output relationship is realized by the transduction in Figure 3.

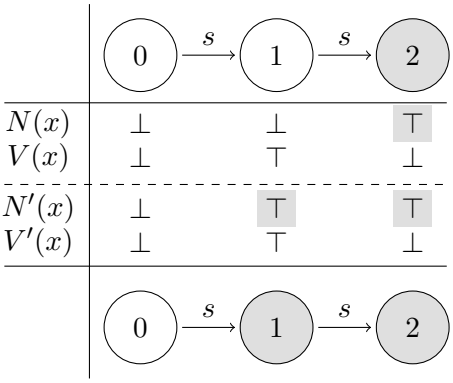


Figure 3: Representation of vowel nasalization as a logic transduction over words with features N and V. The highlighted nodes in the input and output words represent the [+N] elements of the words.

In its full generality, modal logic operates over relational structures called *labeled transition systems* which are very similar to words with two main distinctions that we highlight here. First, the relational structures have relations between nodes

[2]This formula does not account for the edges of words (i.e. when $\mathrm{s}(x)$ is not defined). More precisely, we should set

$$N'(x) = \textbf{if } V(x) \textbf{ then } (\textbf{if } \max(x) \textbf{ then } N(x) \textbf{ else } N(\mathrm{s}(x))) \textbf{ else } N(x),$$

where $\max(x) = \top$ if and only if x is the final index of a word.

(called *accessibility relations*) rather than successor and predecessor functions. Second, we use *modalities* to refer to neighboring indices. For example, given a model with accessibility relation R, the modal expression $\Diamond P$ evaluates to $\top$ at some node x if and only if there is some node y such that $R(x, y)$ and P holds at y. If the relation R is the graph of the successor function (i.e. $R = \{(x, y) \mid y = \mathrm{s}(x)\}$) then $\Diamond P$ evaluates to $\top$ at some index x of a word if and only if $P(\mathrm{s}(x))$ holds. With this equivalence in mind, the BMRS formulas in (3) can be equivalently expressed as the modal formulas in (4), which we refer to as a *modal transduction*. Note that the formulas in (4) do not have variables because the $\Diamond$ modality makes it possible to refer to adjacent indices without a variable.[3]

(4) Modal representation of vowel nasalization

$$V' = V$$
$$N' = (V \wedge \Diamond N) \vee N$$

We consider now an *iterative* nasalization process. In (2), a single vowel becomes nasalized when it is followed by a nasal; no other sounds in a word are affected. (5) presents a vowel nasalization process from Warao in which vowels become nasalized when they are preceded by a nasal sound, and this process iterates through a word until a voiceless stop suspends the nasalization process.

(5) Nasal Spreading in Warao (Osborn, 1966)

a.	esoha-ya	'He pours'
b.	**nãõ-ỹã**	'He saw'
c.	**nãõ**-te	'He will come'
d.	ho**nĩw̃ã**ku-hae	'It is a turtle'
e.	pa**nãpanã-h̃ãẽ**	'It is a porpoise'

The **BMRS** representation of the nasal spreading process in Warao is presented in (6), where the predicate T is a shorthand for voiceless stops, and $\min$ is the predicate which is true only at index 0. Notably, the output predicate N' is *recursive*.

(6) BMRS representation of nasal spreading

$$N'(x) = \begin{array}{l} \textbf{if } N(x) \textbf{ then } \top \textbf{ else} \\ \textbf{if } T(x) \textbf{ then } \bot \textbf{ else} \\ \textbf{if } \min(x) \textbf{ then } \bot \textbf{ else} \\ \mathrm{N}'(px) \end{array}$$

[3]The modal formulas also use the fact that conjunction and disjunction can be expressed as `if...then...else` expressions. For example, $p \wedge q$ can be rewritten as `if` p `then` q `else` $\bot$.

Figure 4 shows the evaluation of the recursive predicate N' for the input-output pair of words (/naote/, [nãõte]). This predicate is evaluated as follows. At index 0, N' evaluates to $\top$ because N holds at index 0. At index 3, N' evaluates to $\bot$ because T holds at index 3. For the remaining indices, $N'(x)$ cannot be evaluated to $\top$ or $\bot$ because $\mathrm{N}'(px)$ does not have a valuation. However, a second run through the `if...then...else` expression yields valuations for indices 1 and 4 because $\mathrm{N}'(0)$ and $\mathrm{N}'(3)$ have valuations. At a third run of the expression, we also get a valuation for index 2. Beyond this point, further iterations of $N'(x)$ will not yield any changes; the final row in Figure 4 is the *least fixed point* of $N'(x)$.

input	n	a	o	t	e
index	0	1	2	3	4
predicates	N	V	V	T	V
$\mathrm{N}'(x)$	$\top$			$\bot$	
$\mathrm{N}'(x)$	$\top$	$\top$		$\bot$	$\bot$
$\mathrm{N}'(x)$	$\top$	$\top$	$\top$	$\bot$	$\bot$
output	n	ã	õ	t	e

Figure 4: Evaluation of the recursive predicate $N'(x)$.

The modal formula which expresses the recursive predicate in (6) is presented in (7). This formula uses the backward modality $\blacklozenge$, where $\blacklozenge P$ evaluates to $\top$ at some node y if and only if there is some node x such that $R(x, y)$ and P holds at x. Intuitively, the backward modality is the analog of the predecessor function over words. The recursive aspect of the formula is handled by the modal operator μX which scopes over $N \vee (\neg T \wedge \blacklozenge X)$ and binds the variable X. The evaluation of this formula is analogous to the evaluation of the recursive BMRS formula in the following sense: the expression $N \vee (\neg T \wedge \blacklozenge X)$ gets re-evaluated iteratively until reaching a fixed point.

(7) Modal representation of nasal spreading

$$N' = \mu X.(N \vee (\neg T \wedge \blacklozenge X))$$

Figure 5 shows the evaluation of the modal μ-calculus expression in (7). We start with X being false at all indices, and evaluate the formula $N \vee (\neg T \wedge \blacklozenge X)$. This formula is true at index 0 (because N is true at 0), and false everywhere else. This becomes the new semantics for X. In other words, X becomes true at index 0 and remains false everywhere else, as shown in the second row of the table. We then evaluate the for-

input	n	a	o	t	e
index	0	1	2	3	4
predicates	N	V	V	T	V
X	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
X	$\top$	$\bot$	$\bot$	$\bot$	$\bot$
X	$\top$	$\top$	$\bot$	$\bot$	$\bot$
X	$\top$	$\top$	$\top$	$\bot$	$\bot$
output	n	ã	õ	t	e

Figure 5: Evaluation of the μX operator.

mula $N \vee (\neg T \wedge \blacklozenge X)$ again with the new truth values for X. This time, $\blacklozenge X$ holds at index 1 and therefore $N \vee (\neg T \wedge \blacklozenge X)$ is true at index 1. The evaluation of X changes, as shown in row 2, with the addition of a $\top$ evaluation for X at index 1. The process repeats again, and now the formula $N \vee (\neg T \wedge \blacklozenge X)$ is true at index 2, and so X becomes true at index 2. In this way, the set of indices where X is true grows with each iteration until reaching a fixed point, where further evaluations of $N \vee (\neg T \wedge \blacklozenge X)$ will not yield any new indices where X holds. The key observation is that the least fixed point of the modal formula in (7) and the least fixed point of the BMRS formula in (6) evaluate to exactly the same thing.

The natural relationship between BMRS and the modal μ-calculus is reminiscent of early work from Chandlee and Jardine (2019), who use a quantifier-free logic equipped with a least fixed point operator to model phonological maps. The modal logic perspective is yet another quantifier-free perspective on the same idea.

Ultimately, this section showed three formal representations of phonological maps. The examples here, though informal, point to an intuitive equivalence between **BMRS** formulas and modal μ-calculus formulas when we restrict our structures to words. Intuitively, we see that references to the successor and predecessor functions can instead be expressed using forward and backward modalities, while recursion in **BMRS** is captured by applications of the fixed point operator.

3 Boolean Monadic Recursive Schemes

To define **BMRS** formally, we fix a single *index variable* x and a countably infinite set $\mathcal{F}$ of unary predicate symbols. **BMRS** terms T and expressions E are defined as in the following recursive

$$\frac{}{\mathbf{w}, i \vdash x \to i} \qquad \frac{\mathbf{w}, i \vdash T \to v}{\mathbf{w}, i \vdash \mathrm{p}(T) \to v - 1} \text{ if } v > 0$$

$$\frac{\mathbf{w}, i \vdash T \to v}{\mathbf{w}, i \vdash \mathrm{s}(T) \to v + 1} \text{ if } v < |w| - 1$$

Figure 6: Derivation system for denotations of terms. The judgment $\mathbf{w}, i \vdash T \to v$ means that term T denotes index v when evaluated at index i in word $\mathbf{w}$.

grammar, where $f \in \mathcal{F}$.

$$\begin{aligned} T &::= x \mid \mathrm{p}(T) \mid \mathrm{s}(T) \\ E &::= \top \mid \bot \mid \min(T) \mid \max(T) \mid \\ &\quad\; f(T) \mid \textbf{if } E \textbf{ then } E \textbf{ else } E \end{aligned}$$

A **BMRS** *program* P has the form

$$\begin{aligned} f_1(x) &:= E_1 \\ &\vdots \\ f_n(x) &:= E_n, \end{aligned}$$

where $f_1, \ldots, f_n \in \mathcal{F}$ are pairwise distinct and $E_1, \ldots, E_n$ are **BMRS** expressions. We refer to $f_j(x) := E_j$ as a *rule* of the program; a *rule head* for the program is an element of $\{f_1, \ldots, f_n\}$. The *input signature*[4] of P is the set $\mathrm{sig}(P)$ containing all $f \in \mathcal{F}$ which appear in some E_j but which do not appear in $\{f_1, \ldots, f_n\}$.

Terms are *index-valued*; a term T in a word $\mathbf{w}$ at index i *denotes* the index j if the judgment $\mathbf{w}, i \vdash T \to j$ is provable according to the derivation system given in Figure 6. Expressions of a **BMRS** program are *Boolean*-valued. An expression occurring in a program P may be evaluated at an index in a $\mathrm{sig}(P)$-word according to the derivation system in Figure 7. The judgment $\mathbf{w}, i \vdash_P E \to b$ indicates that expression E evaluates to $b \in \{\textbf{true}, \textbf{false}\}$ in $\mathrm{sig}(P)$-word $\mathbf{w}$ at index i under program P.

Note that evaluating an expression depends on the program, while evaluating a term does not. One reason for this is the *unfolding rule* for rule heads f_j, which depends on the rule of the form $f_j := E_j$ in program P (if it exists):

$$\frac{\mathbf{w}, i \vdash T \to v \qquad \mathbf{w}, v \vdash_P E_j \to b}{\mathbf{w}, i \vdash_P f_j(T) \to b}$$

[4] Earlier work on **BMRS** distinguishes explicitly between the input and output signatures; here, we *infer* these from the "free symbols" occurring in the program.

Another reason is the *lookup rule* for elements of $\mathrm{sig}(P)$, which requires that the unary predicate σ being evaluated at a term T is defined in the word:

$$\frac{\mathbf{w}, i \vdash T \to v}{\mathbf{w}, i \vdash_P \sigma(T) \to \mathbf{true}} \; (v \in \sigma^{\mathbf{w}})$$

Avoiding confusion over whether the unfolding or lookup rule can be applied to a given unary predicate is why an expression E in program P can *only* be evaluated in a $\mathrm{sig}(P)$-word.

It is worth noting that, by the rules in Figure 6, not every term has a denotation. For example, if i is the largest index in $\mathbf{w}$, then no judgment of the form $\mathbf{w}, i \vdash \mathrm{s}(x) \to j$ is derivable. Similarly, expressions are also not always guaranteed to evaluate; some programs P contain expressions E for which neither $\mathbf{w}, i \vdash_P E \to \mathbf{true}$ nor $\mathbf{w}, i \vdash_P E \to \mathbf{false}$ is derivable according to the rules in Figure 7. As an example, observe that in the program containing only the rules $f(x) := g(x)$ and $g(x) := f(x)$, both $f(x)$ and $g(x)$ do not evaluate at any index in any word. Intuitively, this is because, as f and g are both rule heads (i.e., not in the signature of the program), only the unfolding rule applies. Attempting to construct a proof of some judgment of the form $\mathbf{w}, i \vdash_P f(x) \to b$ leads to an infinite unfolding. In other words, this is a "non-halting" program.

An expression E is *defined* at index i in $\mathbf{w}$ with respect to P if $\mathbf{w}, i \vdash_P E \to b$ for some $b \in \{\mathbf{true}, \mathbf{false}\}$. A **BMRS** program P is *total* on words (resp. strings) if, for every $\mathrm{sig}(P)$-word (resp. string) $\mathbf{w}$, every rule head f of P, and every index $i \in w$, the expression $f(x)$ is defined at i in $\mathbf{w}$ with respect to P.

A **BMRS** *interpretation* is a pair (P, Γ), where Γ is a subset of the rule heads of the program, called the *output signature*. The *input signature* of the interpretation is just the input signature of the program P.

An interpretation (P, Γ) induces a logical transduction $[\![P]\!]_\Gamma : \mathbf{Words}_{\mathrm{sig}(P)} \to \mathbf{Words}_\Gamma$ where $[\![P]\!]_\Gamma(\mathbf{w})$ is the Γ-word with domain w, successor function s^w, and predecessor function p^w, such that

$$\gamma^{[\![P]\!]_\Gamma(\mathbf{w})} := \{i \in w \mid \mathbf{w}, i \vdash_P \gamma(x) \to \mathbf{true}\}$$

for each $\gamma \in \Gamma$. For brevity, we omit the notion of *copy sets* used in (Courcelle, 1994; Courcelle and Engelfriet, 2012; Bhaskar et al., 2020, 2023), which permit interpretations to stretch the length of the input word. However, all results in this paper can be extended to the more general case.

It is shown in (Bhaskar et al., 2023, Theorem 6) that (1) the logical transductions induced by well-defined order-preserving[5] **BMRS** interpretations are rational functions, and (2) every rational function is induced by some well-defined order-preserving **BMRS** interpretation. Fact (1) follows from the fact that order-preserving **BMRS** interpretations can easily be simulated by order-preserving monadic second-order logic (**MSO**) interpretations, which are known to express exactly the rational functions (Filiot, 2015, Theorem 4). Fact (2) is more difficult to show; in (Bhaskar et al., 2023), the authors proceed by solving the *syntactic composition problem* for order-preserving **BMRS** interpretations. The syntactic composition problem is interesting in its own right, but its solution does not yield the simplest proof of fact (2). In Section 5, we show that formulas of the modal μ-calculus express exactly the unary properties expressible by total **BMRS** programs. Since the order-preserving interpretations considered here are determined by their output predicates, this expressive equivalence result constitutes an alternative proof of fact (2).

4 Modal Logic

The *basic modal language* (**ML**) is an extension of propositional logic with *modalities* (Blackburn et al., 2001). In the context of modal logic, unary predicate symbols like those in $\mathcal{F}$ are called *proposition letters* or *variables*. For brevity, rather than describing the formal semantics of modal logic over arbitrary labeled transition systems, as was alluded to in Section 2, we instead interpret modal formulas directly over words. Formulas are evaluated at indices, or equivalently at *pointed words* $(\mathbf{w}, i)$. The semantics of atomic predicates, $\min$, $\max$, conjunction, disjunction, and negation are defined in the natural way, and we set

$$\begin{aligned} \mathbf{w}, i \models \Diamond\varphi \iff\; & s^w(i) \text{ is defined and} \\ & \mathbf{w}, s^w(i) \models \varphi; \\ \mathbf{w}, i \models \blacklozenge\varphi \iff\; & p^w(i) \text{ is defined and} \\ & \mathbf{w}, p^w(i) \models \varphi. \end{aligned}$$

The modal μ-calculus (Kozen, 1983) extends modal logic with fixed point operators, with for-

[5] The interpretations defined in the present paper are also *order-preserving*, as in (Bhaskar et al., 2023).

$$\frac{}{\mathbf{w}, i \vdash_P \top \to \mathbf{true}} \qquad \frac{}{\mathbf{w}, i \vdash_P \bot \to \mathbf{false}} \qquad \frac{\mathbf{w}, i \vdash T \to 0}{\mathbf{w}, i \vdash_P \min(T) \to \mathbf{true}} \qquad \frac{\mathbf{w}, i \vdash T \to v}{\mathbf{w}, i \vdash_P \min(T) \to \mathbf{false}}\ (v > 0)$$

$$\frac{\mathbf{w}, i \vdash T \to |w| - 1}{\mathbf{w}, i \vdash_P \max(T) \to \mathbf{true}} \qquad \frac{\mathbf{w}, i \vdash T \to v}{\mathbf{w}, i \vdash_P \max(T) \to \mathbf{false}}\ (v < |w| - 1) \qquad \frac{\mathbf{w}, i \vdash T \to v}{\mathbf{w}, i \vdash_P \sigma(T) \to \mathbf{true}}\ (v \in \sigma^{\mathbf{w}})$$

$$\frac{\mathbf{w}, i \vdash T \to v}{\mathbf{w}, i \vdash_P \sigma(T) \to \mathbf{false}}\ (v \notin \sigma^{\mathbf{w}}) \qquad \frac{\mathbf{w}, i \vdash T \to v \qquad \mathbf{w}, v \vdash_P E_j \to b}{\mathbf{w}, i \vdash_P f_j(T) \to b}\ (1 \le j \le n)$$

$$\frac{\mathbf{w}, i \vdash_P E \to \mathbf{true} \qquad \mathbf{w}, i \vdash_P E' \to b}{\mathbf{w}, i \vdash_P \mathbf{if}\ E\ \mathbf{then}\ E'\ \mathbf{else}\ E'' \to b} \qquad \frac{\mathbf{w}, i \vdash_P E \to \mathbf{false} \qquad \mathbf{w}, i \vdash_P E'' \to b}{\mathbf{w}, i \vdash_P \mathbf{if}\ E\ \mathbf{then}\ E'\ \mathbf{else}\ E'' \to b}$$

Figure 7: Semantics for **BMRS** expressions over a program P. The judgment $\mathbf{w}, i \vdash_P E \to b$ means expression E evaluates to $b \in \{\mathbf{true}, \mathbf{false}\}$ at index i in word $\mathbf{w}$ under P.

mulas generated by the recursive grammar

$$\varphi := \alpha \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid \nabla\varphi \mid \mu X.\varphi,$$

where $\alpha \in \mathcal{F} \cup \{\min, \max, \top, \bot\}$ and $\nabla \in \{\Diamond, \blacklozenge\}$. In formulas of the form $\mu X.\varphi$, we require X to occur only positively in φ, that is, under the scope of an even number of negation symbols. Given a word $\mathbf{w}$ with domain w and a set $U \subseteq w$, let $\mathbf{w}[X \mapsto U]$ be the word obtained from $\mathbf{w}$ by interpreting X as U. The formula $\mu X.\varphi$ induces an operator $F^{\varphi}_{\mathbf{w},X} : \mathcal{P}(w) \to \mathcal{P}(w)$ by

$$F^{\varphi}_{\mathbf{w},X}(U) = \{i \in w \mid \mathbf{w}[X \mapsto U], i \models \varphi\}.$$

By positivity, $F^{\varphi}_{\mathbf{w},X}$ is monotone, and so it has a least fixed point $\mathrm{lfp}(F^{\varphi}_{\mathbf{w},X})$ by the Knaster-Tarski theorem (Knaster, 1928; Tarski, 1955). We define

$$\mathbf{w}, i \models \mu X.\varphi \iff i \in \mathrm{lfp}(F^{\varphi}_{\mathbf{w},X}).$$

We use an equivalent vectorial presentation for our proofs in Section 5. A vectorial formula χ is a finite system of equations

$$X_1 = \theta_1, \quad \ldots, \quad X_n = \theta_n$$

together with a *designated variable* $X_o \in \{X_1, \ldots, X_n\}$. We refer to the X_i as *recursive variables* and assume that they are pairwise distinct. The right-hand side of each equation is generated by the recursive grammar

$$\theta ::= \alpha \mid \theta \wedge \theta \mid \theta \vee \theta \mid \neg\theta \mid \nabla\theta$$

where $\alpha \in \mathcal{F} \cup \{\min, \max, \top, \bot\}$ and $\nabla \in \{\Diamond, \blacklozenge\}$. Elements of $\mathcal{F} \setminus \{X_1, \ldots, X_n\}$ in the right-hand side of some equation in χ are *free*, and the set of free variables of χ is denoted by $\mathrm{sig}(\chi)$. As above, recursive variables must occur only positively in the right-hand sides which define them.

Let $\mathbf{w}$ be a $\mathrm{sig}(\chi)$-word with domain w, and let $\vec{X}$ denote the tuple $X_1, \ldots, X_n$. For tuples $\vec{U} = (U_1, \ldots, U_n) \in (\mathcal{P}(w))^n$, we write $\mathbf{w}[\vec{X} \mapsto \vec{U}]$ for the $(\mathrm{sig}(\chi) \cup \{X_1, \ldots, X_n\})$-word obtained from $\mathbf{w}$ by interpreting each X_j as U_j. The system χ induces a monotone operator $F^{\chi}_{\mathbf{w}} : (\mathcal{P}(w))^n \to (\mathcal{P}(w))^n$ by setting

$$[\![\theta_j]\!]_{\mathbf{w},\vec{U}} = \{i \in w \mid \mathbf{w}[\vec{X} \mapsto \vec{U}], i \models \theta_j\}, \quad \text{and}$$
$$F^{\chi}_{\mathbf{w}}(\vec{U}) = \left([\![\theta_1]\!]_{\mathbf{w},\vec{U}}, \ldots, [\![\theta_n]\!]_{\mathbf{w},\vec{U}}\right),$$

where the satisfaction relation on pointed words in the definition of $[\![\theta_j]\!]_{\mathbf{w},\vec{U}}$ is ordinary modal satisfaction over the expanded word, with $\min$ and $\max$ interpreted in the natural way. Since recursive variables occur only positively in the right-hand sides of equations, $F^{\chi}_{\mathbf{w}}$ is monotone. Thus, by the Knaster-Tarski theorem (Knaster, 1928; Tarski, 1955), it has a least fixed point

$$\mathrm{lfp}(F^{\chi}_{\mathbf{w}}) = (U^*_1, \ldots, U^*_n).$$

Then, for each recursive variable X_j, we write $\mathbf{w}, i \models_{\chi} X_j$ if and only if $i \in U^*_j$. The formula χ itself is satisfied at i if and only if its designated variable is satisfied there:

$$\mathbf{w}, i \models \chi \iff \mathbf{w}, i \models_{\chi} X_o.$$

We write $\mu\mathbf{ML}^{\mathbf{f}}$ for the full vectorial modal μ-calculus. We write $\mu\mathbf{ML}^{\mathbf{f}}_{+}$ for the syntactic fragment of $\mu\mathbf{ML}^{\mathbf{f}}$ which omits negation. Note that, by Bekić's theorem (Bekić, 1984), the ordinary nested syntax for the μ-calculus is equivalent to the vectorial μ-calculus. Thus our choice of syntax is only made for ease of proof, and does not impact our results.

5 Translating μ-calculus to BMRS

In this section, we prove the main containment needed for our equivalence result: every $\mu\mathbf{ML}^{\mathbf{f}}$ formula can be expressed by a **BMRS** program that halts on every input. The proof first works over strings. We establish an automata-theoretic normal form for $\mu\mathbf{ML}^{\mathbf{f}}_{+}$ and then translate the resulting *directed* formulas into total **BMRS** programs. Together with the containment of total **BMRS** predicates in **MSO** and the equivalence of $\mu\mathbf{ML}^{\mathbf{f}}_{+}$, $\mu\mathbf{ML}^{\mathbf{f}}$, and **MSO** on strings (Arenas et al., 2011, Corollary 3.3), this yields the desired equivalence. The passage from equivalence over strings to equivalence over words is obtained by the standard powerset-alphabet reduction, yielding the following main theorem.

Theorem 2. *For every $\mu\mathbf{ML}^{\mathbf{f}}$ formula φ, there exists a total* **BMRS** *program P with output predicate $X_\varphi \in \mathcal{F}$ such that, for all $sig(\varphi)$-words $\boldsymbol{w}$ and indices $i \in w$, we have*

$$\boldsymbol{w}, i \models \varphi \quad \Longleftrightarrow \quad \boldsymbol{w}, i \vdash_P X_\varphi(x) \to \textbf{\textit{true}}.$$

We first prove the string-level containment of $\mu\mathbf{ML}^{\mathbf{f}}_{+}$ in total **BMRS** by passing through a *directed* normal form for $\mu\mathbf{ML}^{\mathbf{f}}_{+}$ formulas, which we now define.

The *dependency graph* of $(\langle X_i = \theta_i \rangle_{i \le n}, X_o)$ has nodes $X_1, \ldots, X_n$ and an edge $X_r \to X_s$ if and only if X_s occurs in θ_r. An SCC C is *backward* if, for every $X_i \in C$, every occurrence in θ_i of a variable from C is under some backward modality and under no forward modality; C is *forward* if the analogous condition holds with forward and backward reversed.

Definition 3 (Directed vectorial formulas)**.** *A vectorial formula is* directed *if every SCC of its dependency graph is backward or forward. We write* **dir-**$\mu\mathbf{ML}^{\mathbf{f}}_{+}$ *for the class of negation-free directed vectorial formulas.*

Lemma 4. *For every $\mu\mathbf{ML}^{\mathbf{f}}_{+}$ formula φ, there exists a* **dir-**$\mu\mathbf{ML}^{\mathbf{f}}_{+}$ *formula ψ_φ with $sig(\varphi) = sig(\psi_\varphi)$ and, for all $sig(\varphi)$-strings $\boldsymbol{w}$ and indices $i \in w$, we have*

$$\boldsymbol{w}, i \models \varphi \quad \textit{if and only if} \quad \boldsymbol{w}, i \models \psi_\varphi.$$

Proof. Let φ be a $\mu\mathbf{ML}^{\mathbf{f}}_{+}$ formula with $\mathrm{sig}(\varphi) = A$. Let $A^\bullet$ be the marked alphabet which contains, for each $a \in A$, both a and a marked copy $a^\bullet$ (Filiot et al., 2019, Sec. 6.1.3 and Ex. 6.10). Given a pointed A-string $(\mathbf{w}, i)$, let $\mathbf{w}^{\bullet i}$ be the corresponding $A^\bullet$-string obtained by marking the i-th symbol. By the equivalence of $\mu\mathbf{ML}^{\mathbf{f}}_{+}$ and **MSO** on strings (Arenas et al., 2011, Corollary 3.3) and the Büchi-Elgot-Trakhtenbrot theorem (Büchi, 1960), there is a DFA $\mathcal{A} = (Q, q_0, \delta, F)$ over $A^\bullet$ such that $\mathbf{w}, i \models \varphi$ if and only if $\widehat{\delta}(q_0, \mathbf{w}^{\bullet i}) \in F$, where $\widehat{\delta}$ denotes the usual extension of δ to strings.

Using the transition function of $\mathcal{A}$, form a vectorial formula ψ_φ with variables L_q and R_q for each $q \in Q$, and designated variable X_φ. The L_q equations use only the backward modality and record the state of $\mathcal{A}$ before the current position; the R_q equations use only the forward modality and record whether the suffix beginning at the current position is accepted from state q; and the equation for X_φ combines these summaries with the current marked symbol. We omit the routine finite disjunctions spelling out these equations from δ. By construction, $\widehat{\delta}(q_0, \mathbf{w}^{\bullet i}) \in F$ if and only if $\mathbf{w}, i \models \psi_\varphi$. Thus we conclude that $\mathbf{w}, i \models \psi_\varphi$ if and only if $\mathbf{w}, i \models \varphi$.

Clearly ψ_φ is negation-free and $\mathrm{sig}(\psi_\varphi) = A$. Every SCC is contained among the L_q's, among the R_q's, or is $\{X_\varphi\}$; these are backward, forward, and vacuous, respectively. Hence $\psi_\varphi \in$ **dir-**$\mu\mathbf{ML}^{\mathbf{f}}_{+}$. □

Definition 5 (Agreement)**.** *Let P be a* **BMRS** *program, let E be a* **BMRS** *expression, let φ be a $\mu\mathbf{ML}^{\mathbf{f}}$ formula, let $\boldsymbol{w}$ be a $sig(\varphi)$-word, and let $i \in w$. We say that E* agrees with *φ at i in $\boldsymbol{w}$ if E is defined at i in $\boldsymbol{w}$ with respect to P, and $\boldsymbol{w}, i \models \varphi$ if and only if $\boldsymbol{w}, i \vdash_P E \to \textbf{\textit{true}}$.*

Definition 6. *Given a formula $\chi \in$* **dir-**$\mu\mathbf{ML}^{\mathbf{f}}_{+}$ *of the form $(\langle X_i = \theta_i \rangle_{i \le n}, X_o)$, for each right-hand side formula φ, define a* **BMRS** *expression $\mathrm{tr}(\varphi)$ with free variable x by*

$$\begin{aligned}
\mathrm{tr}(b) &:= b, \qquad (b \in \{\top, \bot\}),\\
\mathrm{tr}(X) &:= X(x), \qquad (X \in \mathcal{F}),\\
\mathrm{tr}(\min) &:= \min(x), \qquad \mathrm{tr}(\max) := \max(x),\\
\mathrm{tr}(\varphi \wedge \psi) &:= \textbf{\textit{if}}\ \mathrm{tr}(\varphi)\ \textbf{\textit{then}}\ \mathrm{tr}(\psi)\ \textbf{\textit{else}}\ \bot,\\
\mathrm{tr}(\varphi \vee \psi) &:= \textbf{\textit{if}}\ \mathrm{tr}(\varphi)\ \textbf{\textit{then}}\ \top\ \textbf{\textit{else}}\ \mathrm{tr}(\psi),\\
\mathrm{tr}(\Diamond\varphi) &:= \textbf{\textit{if}}\ \max(x)\ \textbf{\textit{then}}\ \bot\ \textbf{\textit{else}}\ \mathrm{tr}(\varphi)[s(x)],\\
\mathrm{tr}(\blacklozenge\varphi) &:= \textbf{\textit{if}}\ \min(x)\ \textbf{\textit{then}}\ \bot\ \textbf{\textit{else}}\ \mathrm{tr}(\varphi)[p(x)].
\end{aligned}$$

Here, $E[T]$ denotes substitution of the **BMRS** *term T for the free variable x in E. The translated* **BMRS** *program P_χ has output predicate X_o and contains the rule $X_j(x) := \mathrm{tr}(\theta_j)$ for*

each equation $X_j = \theta_j$. Thus the rule heads of P_χ are exactly the recursive variables of χ and $sig(P_\chi) = sig(\chi)$.

Remark 7 (Compositionality of the translation)**.** *The following fact is immediate by induction on φ. Let χ be as above, let P_χ be its* **BMRS** *translation, let* ***w*** *be a $sig(\chi)$-string, and let $i \in w$. Suppose the following atomic correspondence for $\mathrm{tr}(\varphi)$ holds: whenever $Y(T)$ occurs in $\mathrm{tr}(\varphi)$, where Y is a rule head of P_χ and $\boldsymbol{w}, i \vdash T \to h$, then the expression $Y(T)$ is defined at i in* ***w*** *with respect to P_χ, and*

$$\boldsymbol{w}, i \vdash_{P_\chi} Y(T) \to \textbf{\textit{true}} \quad \iff \quad \boldsymbol{w}, h \models_\chi Y.$$

Then $\mathrm{tr}(\varphi)$ is defined at i in ***w*** *with respect to P_χ, and agrees with φ at i in* ***w****.*

Theorem 8. *For every $\mu\mathbf{ML}^{\mathbf{f}}_{+}$ formula φ, there exists a* **BMRS** *program P that is total on strings and has output predicate $X_\varphi \in \mathcal{F}$ such that, for all $sig(\varphi)$-strings* ***w*** *and indices $i \in w$, we have*

$$\boldsymbol{w}, i \models \varphi \quad \iff \quad \boldsymbol{w}, i \vdash_P X_\varphi(x) \to \textbf{\textit{true}}.$$

Proof. Let $\varphi \in \mu\mathbf{ML}^{\mathbf{f}}_{+}$, and choose by Lemma 4 a formula $\chi = (\langle X_i = \theta_i \rangle_{i \le n}, X_o) \in$ **dir-**$\mu\mathbf{ML}^{\mathbf{f}}_{+}$ which is equivalent to φ on strings. Let P_χ be its **BMRS** translation. Let $C_1, \ldots, C_k$ be the SCCs of the dependency graph of χ, topologically ordered so that if $X \in C_j$ and a recursive variable Y occurs in θ_X, then $Y \in C_\ell$ for some $\ell \le j$.

We prove by strong induction on j that every rule head in $C_1 \cup \cdots \cup C_j$ is defined and agrees with the corresponding recursive variable of χ at every index of every $\mathrm{sig}(\chi)$-string. Fix C_j, and assume inductively that

$$\begin{aligned} &\text{for every } \ell < j, \text{ every } Y \in C_\ell, \\ &\text{every } \mathrm{sig}(\chi)\text{-string } \mathbf{w}, \text{ and every index} \\ &h \in w,\ Y(x) \text{ agrees with } Y \text{ at } h \text{ in } \mathbf{w}. \end{aligned} \tag{IH1}$$

Assume that C_j is backward. Fix a $\mathrm{sig}(\chi)$-string $\mathbf{w}$, and proceed by induction on indices i from left to right. Fix an index i, and assume inductively that

$$\begin{aligned} &\text{for every } m < i \text{ and every } Y \in C_j,\ Y(x) \\ &\text{agrees with } Y \text{ at } m \text{ in } \mathbf{w}. \end{aligned} \tag{IH2}$$

Now fix $X \in C_j$, and write $X = \theta_X$ for its equation in χ. We verify the atomic correspondence hypothesis of Remark 7 for $\mathrm{tr}(\theta_X)$ at i. Let $Y(T)$ be an occurrence in $\mathrm{tr}(\theta_X)$, where Y is a rule head of P_χ, and suppose $\mathbf{w}, i \vdash T \to h$. If $Y \in C_j$, then by directionality of C_j and the definition of tr, $T = \mathrm{p}^r(x)$ for some $r > 0$, so $h < i$. Hence (IH2) gives that $Y(x)$ agrees with Y at h in $\mathbf{w}$. Thus $Y(T)$ is defined at i and $\mathbf{w}, i \vdash_{P_\chi} Y(T) \to$ **true** if and only if $\mathbf{w}, h \models_\chi Y$. If $Y \notin C_j$, then by the ordering of the SCCs we have $Y \in C_\ell$ for some $\ell < j$, and the same conclusion follows from (IH1). Thus the atomic correspondence hypothesis of Remark 7 holds, and therefore $\mathrm{tr}(\theta_X)$ is defined at i in $\mathbf{w}$ with respect to P_χ, and agrees with θ_X at i in $\mathbf{w}$.

Since P_χ contains the rule $X(x) := \mathrm{tr}(\theta_X)$, and $X = \theta_X$ is an equation of χ, it follows that $X(x)$ agrees with X at i in $\mathbf{w}$. This closes the induction on indices for the backward component C_j. The case where C_j is forward is analogous, using induction on $|w| - i + 1$. Thus the induction on j gives totality and agreement between all recursive variables of χ and the corresponding rule head of P_χ at all indices of $\mathbf{w}$. In particular, since χ has designated variable X_o, for every $\mathrm{sig}(\chi)$-string $\mathbf{w}$ and every index $i \in w$, $X_o(x)$ agrees with χ at i in $\mathbf{w}$. Taking $P = P_\chi$ and renaming X_o as X_φ, and using equivalence of χ and φ on strings, we have $\mathbf{w}, i \models \varphi \iff \mathbf{w}, i \vdash_P X_\varphi(x) \to$ **true**. □

Proof of Theorem 2. With respect to the class of strings, Theorem 8 shows that every $\mu\mathbf{ML}^{\mathbf{f}}_{+}$ formula can be expressed by a total **BMRS** program. Conversely, a straightforward encoding shows that every total **BMRS** predicate over strings is definable in **MSO**. By (Arenas et al., 2011, Corollary 3.3), **MSO**, $\mu\mathbf{ML}^{\mathbf{f}}$, and $\mu\mathbf{ML}^{\mathbf{f}}_{+}$ are expressively equivalent over strings. Hence $\mu\mathbf{ML}^{\mathbf{f}}$ and total **BMRS** are expressively equivalent over strings. The passage from strings to words is by the standard powerset-alphabet reduction, whose details we omit. □

6 Concluding Remarks

We have shown that total Boolean Monadic Recursive Schemes are equivalent to the modal μ-calculus on words, yielding an alternative proof that order-preserving **BMRS** interpretations capture the rational functions. Because the learning and model-theoretic properties of modal logic are well studied, this connection may benefit phonology; conversely, the programming-language perspective of **BMRS** may also inform modal logic and its applications.

References

Marcelo Arenas, Pablo Barceló, and Leonid Libkin. 2011. Regular languages of nested words: fixed points, automata, and synchronization. *Theory of Computing Systems*, 49(3):639–670.

Hans Bekić. 1984. Definable operations in general algebras, and the theory of automata and flowcharts. In C. B. Jones, editor, *Programming Languages and Their Definition: H. Bekič (1936–1982)*, pages 30–55. Springer Berlin Heidelberg, Berlin, Heidelberg.

Jean Berstel. 2007. *Transductions and Context-Free Languages*. Vieweg+Teubner Verlag Wiesbaden.

Siddharth Bhaskar, Jane Chandlee, and Adam Jardine. 2023. Rational functions via recursive schemes. *Preprint*, arXiv:2302.03074.

Siddharth Bhaskar, Jane Chandlee, Adam Jardine, and Christopher Oakden. 2020. Boolean monadic recursive schemes as a logical characterization of the subsequential functions. In *International conference on language and automata theory and applications*, pages 157–169. Springer.

Patrick Blackburn, Maarten de Rijke, and Yde Venema. 2001. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press.

Julian Bradfield and Colin Stirling. 2006. *Modal Mu-Calculi*, chapter 12. Elsevier.

J. Richard Büchi. 1960. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6):66–92.

Jane Chandlee. 2014. *Strictly Local Phonological Processes*. Ph.D. thesis, University of Delaware.

Jane Chandlee and Adam Jardine. 2019. Quantifier-free least fixed point functions for phonology. In *Proceedings of the 16th Meeting on the Mathematics of Language (MOL)*, pages 50–62. Association for Computational Linguistics.

Jane Chandlee and Adam Jardine. 2021. Computational universals in linguistic theory: Using recursive programs for phonological analysis. *Language*, 97(3):485–519.

Bruno Courcelle. 1994. Monadic second-order definable graph transductions: a survey. *Theoretical Computer Science*, 126(1):53–75.

Bruno Courcelle and Joost Engelfriet. 2012. Monadic second-order transductions. In *Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach*, chapter 7, pages 505–577. Cambridge University Press.

E. A. Emerson and C. S. Jutla. 1991. Tree automata, mu-calculus and determinacy. In *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science*, SFCS '91, page 368–377, USA. IEEE Computer Society.

Emmanuel Filiot. 2015. Logic-automata connections for transformations. In *Indian conference on logic and its applications*, pages 30–57. Springer.

Emmanuel Filiot, Olivier Gauwin, and Nathan Lhote. 2019. Logical and algebraic characterizations of rational transductions. *Logical Methods in Computer Science*, Volume 15, Issue 4:16.

Thomas Graf. 2010. Logics of phonological reasoning. Master's thesis, University of California, Los Angeles.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. In Larry M. Hyman and Frans Plank, editors, *Phonological Typology*, volume 23 of *Phonology and Phonetics*, pages 126–195. De Gruyter Mouton, Berlin, Boston.

Adam Jardine, Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2014. Very efficient learning of structured classes of subsequential functions from positive data. In *The 12th International Conference on Grammatical Inference*, volume 34, pages 94–108.

Douglas Johnson. 1972. *Formal aspects of phonological description*. De Gruyter.

Ronald M Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

Bronisław Knaster. 1928. Un théorème sur les fonctions d'ensembles. *Annales de la Société Polonaise de Mathématique*, 6:133–134.

Raoul Koudijs. 2022. Learning modal formulas via dualities. Master's thesis, Universiteit van Amsterdam.

Dexter Kozen. 1983. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27(3):333–354. Special Issue Ninth International Colloquium on Automata, Languages and Programming (ICALP) Aarhus, Summer 1982.

José Oncina, Pedro García, and Enrique Vidal. 1993. Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:448–458.

Henry A. Osborn. 1966. Warao i: Phonology and morphophonemics. *International Journal of American Linguistics*, 32(2).

Alfred Tarski. 1955. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5(2):285–309.

Balder ten Cate and Raoul Koudijs. 2024. Characterising modal formulas with examples. *ACM Transactions on Computational Logic*, 25(2).

Thomas Wilke. 2001. Alternating tree automata, parity games, and modal μ-calculus. *Bulletin of The Belgian Mathematical Society*, 8:359–391.

Tatevik Yolyan. 2025a. A framework for learning phonological maps as logical transductions. In *Proceedings of the 18th Meeting on the Mathematics of Language*, page 44–58. Association for Computational Linguistics.

Tatevik Yolyan. 2025b. A logical characterization of weak determinism as simultaneous application. *Journal of Logic, Language and Information*.

Tatevik Yolyan. 2025c. *Phonological Expressivity and Learning via Boolean Monadic Recursive Schemes*. Ph.D. thesis, Rutgers The State University of New Jersey, School of Graduate Studies.

Author Index

Anand, Pranav, 258
Anderson, Carolyn Jane, 313
Arun, Ishita, 59
Azadegan, Malek, 171

Bagartti, Pranab, 353
Baidildinova, Moldir, 442
Bakay, Özge, 343
Bárány, András, 72

Cai, Zhenguang, 464
Carslaw, Iona, 72
Chan, Ariel Shuk Ling, 230
Charlow, Simon, 57
Cho, Youngdong, 369
Comer, Jesse, 549
Coulson, Seana, 487
Cummings, Shawn N., 513

Dai, William, 57
De Santo, Aniello, 191, 219
Dillon, Brian, 280, 343
Do, Youngah, 113, 477
Duan, Xufeng, 464

Erk, Katrin, 280

fackerman@ucsd.edu, fackerman@ucsd.edu, 291
Frank, Robert, 57
Futrell, Richard, 338, 442

Gutierrez-Quintero, Andres, 127

Hale, John T., 411
Hampe, Anton, 542
Han, Qi, 379
Hayden, Matthew, 447
Heinz, Jeffrey, 180, 262
Hoefer, Elizabeth Spaulding, 1
Hong, Stella Eunsoo, 496
Huang, Daniel, 37
Huang, Yuhan, 244
Hunter, Tim, 129
Husain, Samar, 59, 353

Ilie, Izabel, 329

Jaeger, T. Florian, 513
Jang, Yongseok, 100
Jardine, Adam, 304
Jarosz, Gaja, 343

Kalivoda, Ágnes, 291
Kamerath, Michael, 191
Kasenov, Daniar, 147
Kastner, Itamar, 72
Khalifa, Salam, 262
Kodner, Jordan, 262
Kurumada, Chigusa, 513
Kwon, Chloe Dokyung, 369

Lambert, Dakotah, 25
Lamont, Andrew, 329
Li, Han, 180
Li, Jane, 365
Li, Jixing, 244, 464
Li, Lucas Y., 427
Li, Yanting, 230, 338
Liao, Xinyu Leslie, 529
Liu, Zoey, 127
Lu, Yiming, 529
Lynch, Zander, 427

Ma, Zhengwu, 244, 464
Mailhot, Fred, 423
Malouf, Robert, 291
Martin, Fermin Moscoso Del Prado, 138
Martin, James, 1
Mayer, Connor, 442
McCoy, R. Thomas, 57

Neu, Eva, 160, 280, 343

Pagliai, Valeria, 127
Paramore, Jonathan Charles, 47
Patil, Abhinav, 365
Paulson, John, 219
Payne, Sarah, 90
Poschl, Jacob, 230
Prasad, Grusha, 313
Prickett, Brandon, 329
Pulido, Emiliana, 127

Qaddoumi, Abed, 262

Rambow, Owen, 262

Rawlins, Kyle, 365
Rawski, Jonathan, 219
Rodó, Sergio José Salazar, 127

Salhan, Suchir, 138
Schijndel, Marten Van, 379, 427
Scontras, Gregory, 338
Shankwitz, Annika, 15
Sharma, Eashani, 59
Siewert, Paul, 294
Singh, Shivam, 408
Steedman, Mark, 72

Tabas, Alejandro, 529
Tan, Frank Lihui, 113
Turk, Utku, 160, 343

Upadhye, Shiva, 442

Verbil, Allison, 129
Viswanathan, Maya, 57

Wagner, Austin, 442
Wang, Anzi, 313
Wang, Chengcheng, 244
Wang, Chenli, 304
Wang, Yang, 460
Warstadt, Alex, 408
Wu, Hongchen, 37
Wu, Yifan, 379

Xie, Xin, 513, 529

Yao, Ruoqing, 258
Yao, Zhaoqian, 464
Yolyan, Tatevik, 549
Yue, Shisen, 411

Zhang, Shuhao, 477
Zhao, Xingyuan, 487
Zhou, Zhenghao, 57